

Слободанка Ђенић
Јелена Митић ▪ Светлана Штрбац

**Решени задаци
на програмским језицима "С" и "С++"**

Збирка испитних задатака

Висока школа електротехнике и рачунарства
струковних студија
Београд, 2018.

Аутори: Слободанка Ђенић, Јелена Митић, Светлана Штрбац
Рецензенти: Ласло Краус, Зоран Бањац
Лектор: Анђелка Ковачевић
Корице: Јелена Лабус (MASSVision)
Издавач: Висока школа електротехнике и рачунарства
струковних студија у Београду
Штампа: Развојно-истраживачки центар
графичког инжењерства ТМФ, Београд
Тираж: 100
Издање: 3.

ISBN 978 - 86 - 85081 – 67 - 5

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004.432.2C (075.8) (076)
004.432.2C++ (075.8) (076)

ЂЕНИЋ, Слободанка, 1965 -

Решени задаци на програмским језицима
“C” и “C++” : збирка испитних задатака /
Слободанка Ђенић, Јелена Митић, Светлана
Штрбац. - 3. изд. - Београд : Висока школа
електротехнике и рачунарства струковних
студија, 2018 (Београд: Развојно-истраживачки
центар графичког инжењерства ТМФ). – 153 стр. :
илустр. ; 24 cm

Тираж 100. – Библиографија: str. 153.

ISBN 978 - 86 - 85081 - 67 – 5

1. Митић, Јелена, 1966- [аутор] 2. Штрбац,
Светлана 1972- [аутор] а) Програмски језик “C” -
Задаци б) Програмски језик “C++” - Задаци
COBISS.SR-ID 256489484

Предговор

Збирка испитних задатака “Решени задаци на програмским језицима С и С++” намењена је студентима прве и друге године Високе школе електротехнике и рачунарства струковних студија у Београду и свима који су савладали програмски језик С и основе програмског језика С++ а желе да се припреме за решавање програмских задатака које могу срести у пракси.

Збирка садржи велики број програма који користе све основне елементе необјектно-оријентисано пројектованих програма, као што су: основни типови података, оператори, наредбе, низови, структуре и функције. Ова збирка чини комплет са збиркама “Основи програмирања на језику С” и “Програмирање на језику С и основи програмирања на језику С++”, које су такође издања Високе школе електротехнике и рачунарства струковних студија у Београду.

Збирка обухвата пет делова. Прва четири дела збирке садрже елементе програмског језика С. Први део показује употребу основних типова података, оператора, и појединих врста наредби. Други део збирке објашњава рад са нумеричким и знаковним низовима. Трећи део збирке показује како се показивачи, функције, динамичка додела меморије и структуре могу искористити за писање програма који ефикасно користе ресурсе рачунарских система. У четвртом делу збирке приказана су решења програма који се позивају из командне линије, као и програма који раде са датотекама. Пети део садржи решења свих претходно поменутих програмских задатака у необјектно-оријентисано пројектованим програмима на језику С++.

Сваки задатак у овој збирци има јасно дефинисану поставку и решење на одговарајућем програмском језику. Решење садржи изворни код (текст програма) који је написан читљиво и објашњен помоћу бројних коментара. Решења неких задатака (или њихових делова) урађена су на више начина (у више примера), мада није искључено постојање решења различитог од било ког решења приказаног овде. Комбиновање појединих елемената програма као и скраћивање његовог записа урађени су до граница релативно разумљивог текста програма

Без обзира на лако разумевање изворног кода у овој збирци, препоручујемо самостално решавање сваког задатка, као и евентуалне измене и додавање изворног кода, да би исти програми користили код решавања што различитијих програмских задатака.

Садржај

1.	Основни типови података, оператори, наредбе селекције, циклуса и скокова у програмима на језику C	1
2.	Нумерички и знаковни низови, сортирање и претраживање низова у програмима на језику C	37
3.	Функције, показивачи, динамичка додела меморије и структуре у програмима на језику C	83
4.	Аргументи из командне линије и рад са датотекама у програмима на језику C	105
5.	Оператори, наредбе, функције, низови, структуре, динамичка додела меморије, аргументи из командне линије и рад са датотекама у програмима на језику C++	127
	Литература	153

**1. Основни типови података,
оператори, наредбе селекције,
циклуса и скокова у програмима
на језику C**

Задатак 1_1.

Написати програм на језику C који:

- а) црта на екрану таблу као шаховску од $n \times n$ поља ($n \leq 20$) од знака (#, @ или %) и празног знака. Број n и знак програм чита са тастатуре. На пример, за $n=4$ и изабран знак #:

```
# #  
# #  
# #  
# #
```

- б) ако се не зада један од предвиђених знакова онда црта таблу од знака *. На пример, за $n=4$:

```
* *  
* *  
* *  
* *
```

Решење 1_1.

```
#include <stdio.h>           /*biblioteka sa funkcijama koje ispituju tipove znakova*/  
#define MAX 20              /*dozvoljena dimenzija table*/  
  
main()  
{   int i, j, n, c, znak = '*';           /*inicijalizacija znaka za crtanje*/  
  
    printf("\nlzabrati znak: #, @ ili %% \n"); /*znak % duplo, jer je specifican*/  
    c = getchar();  
  
    if(c=='#' || c=='@' || c=='%')  
        znak = c;  
  
    do  
    {   printf("\nBroj polja u jednom redu? ");  
        scanf("%d", &n);  
    }while(n<1 || n>MAX);  
  
    /*crtanje table*/  
    for(i=1; i<=n; i++)  
    {   for(j=1; j<=n; j++)  
        {   if(j%2)  
            putchar((i%2)? znak : ' '); /*u svakoj neparnoj koloni*/  
            else  
                putchar((i%2)? ' ' : znak); /*u svakoj parnoj koloni*/  
            putchar("\n"); /*red: neparan/paran*/  
        }  
    }  
}
```


Задатак 1_2.

Написати програм на језику C који:

- а) црта на екрану таблу димензија $n \times n$ (за n парно и $2 \leq n \leq 20$) од великог и истог малог слова (слово се као мало задаје преко тастатуре). На пример за $n=4$ и изабрано слово z:

ZzZz
zZzZ
ZzZz
zZzZ

- б) ако се уместо малог слова зада неки други знак програм се врти у петљи и чека мало слово за цртање табле.

Решење 1_2.

```
#include <stdio.h>
#define MAX 20          /*dozvoljena dimenzija table*/

main()
{
    int i, j, n, slovo, vslovo, pom;

    do
    {
        printf("\nUnesite dimenziju table (paran broj) ");
        scanf("%d",&n);
    }while (n<2 || n>MAX || n%2);

    do
    {
        printf("\nUnesite malo slovo ");
        getchar();          /*zaostali znak od poziva scanf()*/
        slovo=getchar();    /*zadati znak*/
    }while(slovo<'a' || slovo>'z');

    printf("\n\n");

    vslovo=slovo-('a'-'A');    /*veliko slovo od istog malog*/

    /*crtanje table*/
    for(i=1;i<=n;i++)        /*u svakom redu*/
    {
        for(j=1;j<=n;j+=2)    /*stampanje velikog i istog malog slova*/
        {
            putchar(vslovo);
            putchar(slovo);
        }
        /*zamena veliko/malo slovo i prelaz u novi red*/
        pom=vslovo;
        vslovo=slovo;
        slovo=pom;
        putchar('\n');
    }
    putchar('\n');
```

Задатак 1_3.

Написати програм на језику C који:

- а) црта на екрану ромб на следећи начин: ивица треба да буде исцртана великим словом а унутрашњост попуњена истим малим словом. Преко тастатуре се задаје слово, као и дужина странице n ($3 \leq n \leq 10$). На пример за изабрано слово *o* (*o*) и $n=5$:

```
00000
0ooo0
0ooo0
0ooo0
00000
```

- б) ако се не зада ниједно слово онда црта исту таблу од слова *A* (*a*).

Решење 1_3.

```
#include <stdio.h>
#include <ctype.h>          /*biblioteka sa funkcijama koje ispituju tipove znakova*/
#define MAX 10              /*dozvoljena duzina stranice*/

main()
{
    int n, i, j, znak, slovo = 'a', vslovo = 'A';

    do
    {
        printf("\nUnesite stranicu kvadrata "); scanf("%d",&n);
    }while (n<3 || n>MAX);

    printf("\nUnesite slovo "); getchar(); znak=getchar();
    if(isalpha(znak))
    {
        slovo = tolower(znak);    /*malo slovo od istog velikog*/
        vslovo= toupper(znak);    /*veliko slovo od istog malog*/
    }

    /*crtanje romba*/
    for(i=1;i<=n;i++)            /*u svakom redu*/
    {
        for(j=1;j<=n-i+1;j++)    /*odvajanje praznih mesta*/
            putchar(' ');
        if(i==1 || i==n)          /*u prvom i poslednjem redu*/
            for(j=1;j<=n;j++)
                putchar(vslovo);
        else                      /*u ostalim redovima*/
        {
            putchar(vslovo);
            for(j=1;j<=n-2;j++)    /*mala slova izuzev 1. i poslednjeg*/
                putchar(slovo);
            putchar(vslovo);
        }
        putchar('\n');
    }
    putchar('\n');
```

Задатак 1_4.

Написати програм на језику C који:

- а) црта на екрану ромб на следећи начин: ивица треба да буде исцртана малим словом, а унутрашњост попуњена истим великим словом. Преко тастатуре се задаје слово, као и дужина странице n ($3 \leq n \leq 10$). На пример за изабрано слово *o* (*o*) и $n=5$:

```
ooooo
o000o
o000o
o000o
ooooo
```

- б) ако се не зада ниједно слово онда црта исту таблу од слова *A* (*a*).

Решење 1_4.

```
#include <stdio.h>
#include <ctype.h>
#define MAX 10                                /*dozvoljena duzina stranice*/

main()
{
    int i, j, n, znak, slovo='a', vslovo='A';

    do
    {
        printf("\nUnesite stranicu kvadrata "); scanf("%d",&n);
    }while (n<3 || n>MAX);

    printf("\nUnesite slovo "); getchar(); znak=getchar();

    if(isalpha(znak))
    {
        slovo = tolower(znak);    /*malo slovo od istog velikog*/
        vslovo= toupper(znak);    /*veliko slovo od istog malog*/
    }

    /*crtanje romba*/
    for(i=1;i<=n;i++)                /*u svakom redu*/
    {
        for(j=1;j<=i;j++)            /*odvajanje praznih mesta*/
            putchar(' ');
        if(i==1 || i==n)              /*u prvom i poslednjem redu*/
            for(j=1;j<=n;j++)
                putchar(slovo);
        else                          /*u ostalim redovima*/
        {
            putchar(slovo);
            for(j=1;j<=n-2;j++)
                putchar(vslovo);    /*velika slova izuzev 1. i poslednjeg*/
            putchar(slovo);
        }
        putchar('\n');
    }
    putchar('\n');
```

Задатак 1_5.

Написати програм на језику C који:

- а) црта на екрану троугао у n ($n \leq 10$) редова на следећи начин: сваки непарни ред црта великим словом а парни истим малим словом. Број редова n и слово (велико или мало) програм чита са тастатуре. На пример, за $n=4$ и изабрано слово A (a):

```
  A
aaa
AAAAA
aaaaaaa
```

- б) ако се не зада слово онда црта троугао само од знака *. На пример, за $n=4$:

```
  *
 ***
*****
*****
```

Решење 1_5.

```
#include <stdio.h>
#include <ctype.h>
#define MAX 10                                /*dozvoljen broj redova*/

main()
{
    int i, j, n, c, znak;

    printf("\nslovo: "); c = getchar();

    if(!isalpha(c))                            /*ako se ne zada slovo*/
        c = '*';

    do
    {
        printf("\nBroj redova? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX);

    /*crtanje trougla*/
    printf("\n");
    for(i=1; i<=n; i++)
    {
        if(i%2)                                /*u svakom neparnom redu veliko slovo*/
            znak = toupper(c);
        else                                    /*u svakom parnom redu malo slovo*/
            znak = tolower(c);

        for(j=1; j<=n-i; j++)                  /*odvajanje praznih mesta*/
            putchar(' ');
        for(j=1; j<=2*i-1; j++)                /*stampanje znakova*/
            putchar(znak);
        putchar('\n');                        /*prelaz u novi red trougla*/
    }
    putchar('\n');
}
```

Задатак 1_6.

Написати програм на језику C који:

- а) црта на екрану стрелицу нагоре од знака * (троугао од стрелице у n ($n \leq 10$) редова и остатак од стрелице у m ($m \leq 10$) редова). Бројеве n и m програм чита са тастатуре. На пример, за задате вредности $n=4$ и $m=2$:

```
      *
     ***
    *****
   *****
  *
 *
```

- б) понавља корак под а) све док се задају вредности n и m у траженом опсегу.

Решење 1_6.

```
#include <stdio.h>
#define MAX 10                                /*dozvoljen broj redova*/

main()
{
    int i, j, n, m;

    while(1)
    {
        printf("\nBroj redova u trouglu? ");
        scanf("%d", &n);
        printf("\nBroj ostalih redova? ");
        scanf("%d", &m);

        if(n<1 || n>MAX || m<1 || m>MAX)
            break;

        /*crtanje trougla*/
        for(i=1; i<=n; i++)
        {
            for(j=1; j<=n-i; j++)          /*odvajanje praznih mesta*/
                putchar(' ');
            for(j=1; j<=2*i-1; j++)          /*stampanje znakova*/
                putchar('*');
            putchar('\n');
        }

        /*crtanje ostatka strelice*/
        for(i=1; i<=m; i++)
        {
            for(j=1; j<=n-1; j++)          /*odvajanje praznih mesta*/
                putchar(' ');
            putchar('*');                  /*stampanje jednog znaka*/
            putchar('\n');
        }
    }
}
```

Задатак 1_7.

Написати програм на језику C који:

- а) црта на екрану стрелицу надоле од знака * (троугао од стрелице у n ($n \leq 10$) редова и остатак од стрелице у m ($m \leq 10$) редова). Бројеве n и m програм чита са тастатуре. На пример, за задате вредности $n=4$ и $m=2$:

```
      *
      *
*****
*****
***
*
```

- б) понавља корак под а) све док се задају вредности n и m у траженом опсегу.

Решење 1_7.

```
#include <stdio.h>
#define MAX 10                                /*dozvoljen broj redova*/

main()
{
    int i, j, n, m;

    while(1)
    {
        printf("\nBroj redova u trouglu? ");
        scanf("%d", &n);
        printf("\nBroj ostalih redova? ");
        scanf("%d", &m);

        if(n<1 || n>MAX || m<1 || m>MAX)
            break;

        /*crtanje prvog dela strelice*/
        for(i=1; i<=m; i++)
        {
            for(j=1; j<=n-1; j++)             /*odvajanje praznih mesta*/
                putchar(' ');
            putchar('*');                      /*stampanje jednog znaka*/
            putchar('\n');
        }

        /*crtanje trougla od strelice*/
        for(i=1; i<=n; i++)
        {
            for(j=1; j<=i-1; j++)             /*odvajanje praznih mesta*/
                putchar(' ');
            for(j=1; j<=2*(n-i)+1; j++)        /*stampanje znakova*/
                putchar('*');
            putchar('\n');
        }
    }
}
```

Задатак 1_8.

Написати програм на језику C који:

- а) црта на екрану делтоид на следећи начин: ивица треба да буде исцртана великим словом а унутрашњост попуњена истим малим словом. Преко тастатуре се задају слово и дужина странице n ($2 \leq n \leq 15$). На пример за изабрано слово S (s) и $n=4$:

```
      S
     SsS
    SsssS
   SsssssS
  SsssS
   SsS
    S
```

- б) ако се не зада ниједно слово онда црта исту таблу од слова A (a).

Решење 1_8.

```
#include <stdio.h>
#include <ctype.h>
#define MAX 15                                /*dozvoljena duzina stranice*/

main()
{
    int n, i, j, k, znak, slovo='a', vslovo='A';

    do
    {
        printf("\nUnesite stranicu deltoida ");
        scanf("%d",&n);
    }while (n<2|| n>MAX);

    printf("\nUnesite slovo ");
    getchar();                                /*zaostali znak od poziva scanf()*/
    znak=getchar();                            /*zadati znak*/

    if(isalpha(znak))
    {
        slovo = tolower(znak);
        vslovo= toupper(znak);
    }

    /*crtanje gornje polovine deltoida*/
    printf("\n\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=MAX-i;j++)                /*odvajanje praznih mesta*/
            putchar(' ');
        for(k=1;k<=2*i-1;k++)                /*stampanje znakova*/
        {
            if(k==1 || k==2*i-1)              /*prvi i poslednji*/
                putchar(vslovo);
            else                               /*ostali znakovi*/
                putchar(slovo);
        }
        putchar('\n');
    }
}
```

```

/*crtanje donje polovine deltoida*/
for(i=n-1;i>=1;i--)
{
    for(j=1;j<=MAX-i;j++)
        putchar(' ');
    for(k=1;k<=2*i-1;k++)
        if(k==1 || k==2*i-1)
            putchar(vslovo);
        else
            putchar(slovo);
    putchar('\n');
}
}

```

Задатак 1_9.

Написати програм на језику C који:

- а) црта на екрану јелку (са два троугла и подножјем у два реда) за преко тастатуре задато слово, као и број n ($2 \leq n \leq 10$). На пример за задато слово W и број $n=3$:

```

      W
    WWW
  WWWWWW
    W
  WWW
  WWWWWW
    W
    W

```

- б) понавља корак под а) све док се задаје вредност n у траженом опсегу.

Решење 1_9.

```

#include <stdio.h>
#include <ctype.h>
#define MAX 10 /*dozvoljena duzina stranice*/

main()
{
    int n, i, j, k, slovo;

    while(1)
    {
        printf("\nUnesite dimenziju jelke ");
        scanf("%d",&n);

        if(n<2 || n>MAX)
            break;

        do
        {
            printf("\nUnesite slovo ");
            getchar(); slovo=getchar();
        }while (!isalpha(slovo));
    }
}

```



```

/*crtanje jelke*/
putchar('\n');
for(k=1;k<=2;k++)                /*krošnja jelke od dva trougla*/
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=MAX-i;j++)
            putchar(' ');
        for(j=1;j<2*i;j++)
            putchar(slovo);
        putchar('\n');
    }
for(k=1;k<=2;k++)                /*podnozje jelke u dva reda*/
{
    for(j=1;j<=MAX-1;j++)
        putchar(' ');
    putchar(slovo);
    putchar('\n');
}
putchar('\n');
}

```

Задатак 1_10.

Написати програм на језику C који:

- а) црта на екрану пахуљу димензије n (за непаран број n и $5 \leq n \leq 15$) од знака #, @ или \$. Број n и знак (којим црта пахуљу) програм чита са тастатуре. На пример, за $n=7$ и изабран знак *:

```

* * *
* * *
***
*****
***
* * *
* * *

```

- б) понавља корак под а) све док се задаје вредност n у траженом опсегу.

Решење 1_10.

```

#include <stdio.h>
#define MAX1 5                /*minimalna duzina stranice*/
#define MAX2 15              /*dozvoljena duzina stranice*/

main()
{
    int n, i, j, k, znak;

    while(1)
    {
        printf("\nUnesite dimenziju slova (neparan broj) ");
        scanf("%d",&n);
        if(n<MAX1 || n>MAX2 || (n+1)%2)
            break;
    }
}

```

```

do
{
    printf("\nUnesite znak (*,#,+, -) ");
    getchar(); znak=getchar();
}while (znak!='*' && znak!='#' && znak!='+' && znak!='-');

/*crtanje pahulje*/
for(i=1;i<=n;i++)          /*u svakom redu pahulje*/
{
    for(k=1;k<MAX2/2;k++)  /*pomeranje crteza u desno*/
        putchar(' ');
    for(j=1;j<=n;j++)      /*u svakoj koloni pahulje*/
        if(i==j || i+j==n+1 || i==(n+1)/2 || j==(n+1)/2)
            putchar(znak);
        else
            putchar(' ');
    putchar("\n");
}
}

```

Задатак 1_11.

Написати програм на језику C који:

- а) црта на екрану стрелицу удесно, дебљине n ($2 \leq n \leq 10$), од знака који није слово. Преко тастатуре се задају знак и број n . На пример за изабран знак * и $n=4$ стрелица треба да изгледа на следећи начин:

```

****
****
****
***** (дужина средњег дела је 3*n-1)
****
****
****

```

- б) ако се зада слово онда исписује на екрану одговарајућу поруку.

Решење 1_11.

```

#include <stdio.h>
#include <ctype.h>
#define MAX1 2          /*minimalna duzina stranice*/
#define MAX2 10         /*dozvoljena duzina stranice*/

main()
{
    int n, i, j, znak;

    do
    {
        printf("\nUnesite sirinu strelice ");
        scanf("%i",&n);
    }while (n<MAX1|| n>MAX2);
}

```


Решење 1_12.

```
#include <stdio.h>
#include <ctype.h>
#define MAX 10          /*dozvoljena duzina stranice*/

main()
{
    int n,i,j,k, znak;

    do
    {
        printf("\nUnesite stranicu osmougaonika ");
        scanf("%d",&n);
    }while (n<1 || n>MAX);

    do
    {
        printf("\nUnesite znak razlicit od slova ");
        getchar(); znak=getchar();
    }while (isalpha(znak));

    /*crtanje gornjeg dela osmougaonika*/
    putchar('\n');
    for(i=1;i<=n;i++)
    {
        for (j=1;j<MAX-i;j++)
            putchar(' ');
        for(j=1;j<=n+2*(i-1);j++)
            putchar(znak);
        putchar('\n');
    }

    /*crtanje srednjeg dela osmougaonika*/
    for(k=1;k<=n-2;k++)
    {
        for (j=1;j<MAX-n;j++)
            putchar(' ');
        for(j=1;j<=n+2*(n-1);j++)
            putchar(znak);
        putchar('\n');
    }

    /*crtanje donjeg dela osmougaonika*/
    for(i=n;i>=1;i--)
    {
        for (j=1;j<MAX-i;j++)
            putchar(' ');
        for(j=1;j<=n+2*(i-1);j++)
            putchar(znak);
        putchar('\n');
    }
    putchar('\n');
}
```

Задатак 1_13.

Написати програм на језику C који:

- а) чита са тастатуре целе бројеве n и m ($n \leq 10$ и $m \leq 10$), где је n број редова и m број колона, а онда приказује на екрану матрицу $n \times m$ са редним бројем на почетку сваког реда, која за вредности $n=3$ и $m=3$ има облик:

```
1-11-21-31
2-12-22-32
3-13-23-33
```

- б) понавља корак под а) све док су вредности n и m учитане са тастатуре, у релацији $n \leq m$.

Решење 1_13.

```
#include <stdio.h>
#define MAX 10 /*dozvoljena dimenzija matrice*/

main()
{
    int i, j, n, m;

    while(1)
    {
        do
        {
            printf("\n\nZadati vrednosti n i m");
            printf("\n(za kraj zadati prvu vrednost vecu od druge) ");
            scanf("%d%d", &n, &m);
        }while(n<1 || n>MAX || m<1 || m>MAX);

        if(n>m)
            break;

        for(i=1; i<=n; i++) /*u svakom od n redova*/
        {
            printf("\n%d", i); /*redni broj reda*/
            for(j=1; j<=m; j++) /*m parova vrednosti*/
                printf("-%d%d", j, i);
        }
    }
}
```

Задатак 1_14.

Написати програм на језику C који:

- а) чита са тастатуре цео број n ($n \leq 10$) а онда приказује на екрану матрицу $n \times n$ са редним бројем на почетку сваке колоне, која за вредност $n=3$ има облик:

```
1    2    3
33   23   13
32   22   12
31   21   11
```

- б) понавља корак под а) све док је вредност n у задатом опсегу.

Решење 1_14.

```
#include <stdio.h>
#define MAX 10 /*dozvoljena dimenzija matrice*/

main()
{
    int i, j, n;

    while(1)
    {
        printf("\n\nZadati vrednost n (1<=n<=10) ");
        scanf("%d", &n);

        if(n<1 || n>MAX)
            break;

        /*prikaz matrice*/
        for(i=1; i<=n; i++) /*redni brojevi kolona*/
            printf(" %d", i);
        printf("\n");

        for(i=n; i>=1; i--) /*nxn parova vrednosti*/
        {
            printf("\n");
            for(j=n; j>=1; j--)
                printf(" %d%d", j, i);

        }

    }
}
```

Задатак 1_15.

Написати програм на језику C који:

- чита са тастатуре један знак, само ако је у опсегу штампајућих знакова, и испишује на екрану његову *ASCII* вредност у децималном облику,
- за изабране бројеве *n* и *m*, сваки у опсегу од 3 до 8 (на пример *n*=5, *m*=3), приказује на екрану *n* квадрата, сваки од *m**x**m* изабраних знакова (на пример &), који су одвојени квадратима од *m**x**m* празних знакова:

```
&&&    &&&    &&&    &&&    &&&
&&&    &&&    &&&    &&&    &&&
&&&    &&&    &&&    &&&    &&&
```

- понавља кораке под а) и б) све док се не зада знак тачка (.).

Решење 1_15.

```
#include <stdio.h>
#include <ctype.h>
#define MAX1 3 /*donja granica opsega za brojeve*/
#define MAX2 8 /*gornja granica opsega za brojeve*/
```

```

main()
{
    int i, j, k, n, m, znak;

    while(1)
    {
        do
        {
            printf("Zadati znak: "); znak = getchar(); getchar();
        }while(!isprint(znak));

        if(znak=='.')
            break;

        printf("\nDecimalna vrednost unetog znaka: %d", znak);
        do
        {
            printf("\n\nIzabrati n i m: ");
            scanf("%d%d", &n, &m);
            getchar();
        }while(n<MAX1 || n>MAX2 || m<MAX1 || m>MAX2);

        for(i=1; i<=m; i++)
        {
            for(j=1; j<=n; j++)
            {
                for(k=1; k<=m; k++)
                    putchar(znak);
                for(k=1; k<=m; k++)
                    putchar(' ');
            }
            putchar('\n');
        }
    }
}

```

Задатак 1_16.

Написати програм на језику C који:

а) исписује на екрану заглавље табеле у следећем облику:

10	8	16	znak
----	---	----	------

б) за сваки штампајући знак из *ASCII* табеле, од знака *!* до знака *@*, исписује на екрану у посебном реду: вредност у децималном, окталном и хексадецималном облику, као и сам знак,

в) за изабран знак из ове табеле (на пример *?*) број *n* у опсегу од 3 до 20 (на пример 4) и једну од 4 опције (*U*, *D*, *L* или *R*) даје један од следећих приказа на екрану:

?	?	????	????	????
?	?	?	?	?
?	?	?	?	?
????	?	?	????	????

Решење 1_16.

```
#include <stdio.h>
#define MAX1 3          /*donja granica dozvoljenog opsega*/
#define MAX2 20         /*gornja granica dozvoljenog opsega*/

main()
{
    int i, j, n, znak, opcija;

    /*zaglavlje tabele*/
    printf("\n 10  8 16 znak");
    printf("\n_____ \n\n");

    /*tabela ASCII vrednosti*/
    for(i='!'; i<='@'; i++)
        printf("%5d%5o%5x%5c\n", i, i, i, i);

    do
    {
        printf("\nlzabrati znak iz datog opsega: ");
        znak = getchar(); getchar();
    }while(znak<'!' || znak>'@');

    do
    {
        printf("\nlzabrati opciju (U, D, L ili R): ");
        opcija = getchar(); getchar();
    }while(opcija!='U' && opcija!='D' && opcija!='R' && opcija!='L');

    do
    {
        printf("\lzabrati duzinu strane (od %d do %d): ", MAX1,MAX2);
        scanf("%d", &n);
    }while(n<MAX1 || n>MAX2);

    /*prikaz rezultata na ekranu*/
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)
        {
            if(i==1 && opcija!='U' || i==n && opcija!='D')
                putchar(znak);
            else
                if(j==1 && opcija!='L' || j==n && opcija !='R')
                    putchar(znak);
                else
                    putchar(' ');
        }
        putchar('\n');
    }
}
```


Задатак 1_17.

Написати програм на језику C који:

а) исписује на екрану заглавље табеле у следећем облику:

```
KOD   SLOVO   kod   slovo
```

б) за свако слово из задатог опсега исписује на екрану, у посебном реду табеле: децималну вредност *ASCII* кода великог слова, само велико слово децималну вредност *ASCII* кода истог малог слова, као и исто мало слово,

в) за изабрано слово из ове табеле (на пример *a*), бројеве *n* и *m*, сваки у опсегу од 1 до 10 (на пример *n*=3 и *m*=10), приказује на екрану $2 \cdot n$ редова од по *m* слова. Сваки ред слова има одговарајућу заграду и померен је за хоризонтални табулатор у односу на претходни ред:

```
{aaaaaaaaa
  {aaaaaaaaa
    {aaaaaaaaa
      aaaaaaaaa}
    aaaaaaaaa}
  aaaaaaaaa}
 aaaaaaaaa}
```

Решење 1_17.

```
#include <stdio.h>
#include <ctype.h>
#define MAX1 1                                /*donja granica dozvoljenog opsega*/
#define MAX2 10                              /*gornja granica dozvoljenog opsega*/

main()
{
    int i, j, n, m, slovo, s1, s2, raz='a'-'A';

    do
    {
        printf("\nPocetno slovo: ");s1=getchar();getchar();
        printf("\nPoslednje slovo: ");s2 = getchar();getchar();
    }while(!isalpha(s1) || !isalpha(s2));

    /*tabela ASCII kodova*/
    printf("\n\n kod SLOVO kod slovo\n\n");
    for(i=toupper(s1); i<=toupper(s2); i++)
        printf("%5d%5c%5d%5c\n", i, i, i+raz, i+raz);

    do
    {
        printf("\nlzabрати slovo iz datog opsega: ");
        slovo = getchar();getchar();
    }while(slovo<s1 || slovo>s2);

    do
    {
        printf("\nlzabрати broj redova i broj slova (od 1 do 10): ");
        scanf("%d%d", &n, &m);
    }while(n<MAX1 || n>MAX2 || m<MAX1 || m>MAX2);
```

```

for(i=1; i<=n; i++)
{
    for(j=1; j<i; j++)                /*pomeraј*/
        putchar('\t');
    putchar('{');                    /*zagrada i znakovi*/
    for(j=1; j<=m; j++)
        putchar(slovo);
    putchar('\n');
}
for(i=1; i<=n; i++)
{
    for(j=1; j<n-i+1; j++)            /*pomeraј*/
        putchar('\t');
    for(j=1; j<=m; j++)              /*znakovi i zagrada*/
        putchar(slovo);
    putchar('}');
    putchar('\n');
}
}

```

Задатак 1_18.

Написати програм на језику C који:

- чита са тастатуре цео број n , величине четири бајта, задат у хексадецималном облику,
- одређује и приказује на екрану позицију (од бита најмање тежине) првог и последњег појављивања бинарне комбинације 101 у задатом броју n .

Решење 1_18.

```

#include <stdio.h>
#define VEL_BAJTA      8      /*broj bitova u jednom bajtu*/
#define VEL_MASKE      3      /*broj bitova od maske*/
#define MASKA          5      /*decimalna vrednost kombinacije 101*/

main()
{
    int i, poz_prva=0, poz_posl=0;
    long n;

    printf("\nBroj u heksadecimalnom obliku?");
    scanf("%lx", &n);

    for(i=0; i<4*VEL_BAJTA-VEL_MASKE; i++)
    {
        if((n & MASKA) == MASKA)
        {
            if(!poz_prva)
                poz_prva = i;
            poz_posl = i;
        }
        n >>= 1;
    }
}

```

```
printf("\nPrvo pojavljivanje kombinacije 101: %d", poz_prva);
printf("\nPoslednje pojavljivanje kombinacije 101: %d\n\n", poz_posl);
}
```

Задатак 1_19.

Написати програм на језику C који:

- чита са тастатуре цео број n , величине четири бајта, задат у децималном или хексадецималном облику,
- одређује и приказује на екрану број парова суседних битова задатог броја који имају различите вредности (у бинарном облику: 01 или 10).

Решење 1_19.

```
#include <stdio.h>
#define VEL_BAJTA      8      /*broj bitova u bajtu*/
#define MASKA1         1      /*u binarnom obliku 01*/
#define MASKA2         2      /*u binarnom obliku 10*/
#define MASKA          3      /*u binarnom obliku 11*/

main()
{
    int i, n, broj_razl=0;

    printf("\nBroj u decimalnom ili heksadecimalnom obliku? ");
    scanf("%i", &n);

    for(i=1; i<=2*VEL_BAJTA; i++)
    {
        if((n & MASKA) == MASKA1 || (n & MASKA) == MASKA2)
            broj_razl++;
        n >>= 1;
    }
    printf("Broj parova razlicitih vrednosti: %d\n", broj_razl);
}
```

Задатак 1_20.

Написати програм на језику C који:

- чита са тастатуре цео број n , величине четири бајта, задат у децималном или хексадецималном облику,
- приказује на екрану поруку о томе да ли су сви битови задатог броја уређени тестерасто (имају вредности редом: 0101... или 1010...) или не,
- одређује и приказује на екрану број битова од задатог броја који имају вредност 1, као и број битова између првог и последњег појављивања вредности 1 у бинарном облику задатог броја.

Решење 1_20.

```
#include<stdio.h>
#define VEL_BAJTA      8                /*broj bitova u bajtu*/
#define TESTER1        0xaaaa          /*u bin. obliku: 1010....*/
#define TESTER2        0x5555          /*u bin. obliku: 0101....*/

main()
{   int i,x;
    int jedinica=0, prva=0, posl=0;

    printf("\nHeksadecimalna vrednost broja u opsegu od 2 bajta: ");
    scanf("%x",&x);

    if(x==TESTER1 || x==TESTER2)
    { printf("\nBitovi zadatog broja su testerasto uredjeni.");
      printf("\nU binarnom ima %d jedinica.", VEL_BAJTA);
      printf("\nlzmedju 1. i poslednje jedinice je %d bitova.\n",2*VEL_BAJTA-3);
    }
    else if(x)
    {   for(i=0;i<2*VEL_BAJTA;i++)
        if((x>>i)&1)
        {   jedinica++;
            if(!prva)
            {   prva=i+1;
                posl=i+1;
            }
            else
                posl=i+1;
        }
      printf("\nBitovi zadatog broja nisu testerasto uredjeni.");
      printf("\nU binarnom zapisu ima %d jedinica.",jedinica);
      printf("\nlzmedju prve i poslednje jedinice ima %d bitova.\n",posl-prva-1);
    }
    else
        printf("\nU binarnom zapisu zadatog broja nema jedinica");
}
```

Задатак 1_21.

Написати програм на језику C који:

- чита са тастатуре цео број, величине четири бајта, задат у хексадецималном облику,
- одређује и приказује на екрану позиције првог и последњег појављивања јединице у бинарном облику задатог броја (1. бит је бит најмање тежине),
- мења вредности битовима задатог броја n -тог бита (од бита најмање тежине) и приказује на екрану један по један бајт (од најнижег) од тако формираног броја.

Решење 1_21.

```
#include <stdio.h>
#define VEL_BAJTA      8      /*broj bitova u bajtu*/
#define MASKA          0xff   /*u binarnom obliku sve jedinice*/

main()
{
    int i, n;
    long x, broj, poz_prve=0, poz_posl=0;

    printf("\nBroj u heksadecimalnom obliku?");
    scanf("%lx", &x);

    do
    {
        printf("\nRedni broj bita za izmenu?");
        scanf("%d", &n);
    }while(n<1 || n>4*VEL_BAJTA);

    broj=x;
    /*odredjivanje prve i poslednje jedinice*/
    for(i=1; i<=4*VEL_BAJTA; i++)
    {
        if(broj&1)
        {
            if(!poz_prve)
                poz_prve = i;
            poz_posl = i;
        }
        broj >>= 1;
    }

    printf("\nZadati broj u heksadecimalnom obliku: %#lx", x);
    printf("\nPrvo pojavljivanje jedinice: %d", poz_prve);
    printf("\nPoslednje pojavljivanje jedinice: %d\n", poz_posl);

    /*izmena vrednosti n-tog bita*/
    x ^= 1<<(n-1);          /*koristi operator ^ za izmenu vrednosti*/

    printf("\nSvi bajtovi (od najnižeg) izmenjenog broja:");
    for(i=0; i<4; i++)        /*prikaz jednog po jednog od 4 bajta*/
        printf("\n%d. bajt: %x", i+1, (broj>>(i*VEL_BAJTA))&MASKA);
}
```

Задатак 1_22.

Написати програм на језику C који:

- чита са тастатуре цео број x , величине четири бајта, задат у хексадецималном облику, као и цео број n задат у децималном облику,
- формира број у тако што за прва три бита у задатом броју x (од бита најмање тежине) сачува вредности, а остале битове поставља на нулу,

- в) формира број z тако што за првих n битова у задатом броју x (од бита најмање тежине) сачува вредности, а остале битове поставља на нулу,
- г) приказује на екрану, у бинарном облику, бројеве x , y и z тако што у сваком од ових бројева одваја групе од по четири бита (број 0x8888 приказује на следећи начин: 1000 1000 1000 1000).

Решење 1_22.

```
#include <stdio.h>
#define MASKA1 7 /*vrednost 1 za prva tri bita, od bita najmanje tezine*/
#define BITOVA 16 /*opseg od dva bajta za podatak*/

main()
{
    int i, iter, n, broj, maska2=0, x, y, z;

    printf("x (u heksadecimalnom obliku): ");
    scanf("%x", &x);

    do
    {
        printf("n (u decimalnom obliku): ");
        scanf("%d", &n);
    }while(n<1 || n>BITOVA);

    for(i=0; i<n;i++)
        maska2 |= (1<<i);

    y = x & MASKA1; /*y ima vrednost 1 za prva 3 bita*/
    z = x & maska2; /*z ima vrednost 1 za prvih n bitova*/

    /*prikaz sva tri broja u binarnom obliku*/
    for(iter=1; iter<=3; iter++) /*u svakoj iteraciji*/
    {
        broj=(iter==1)?x:((iter==2)?y:z); /*broj dobija vrednost x,y ili z*/

        for(i=1; i<=BITOVA; i++) /*u opsegu od dva bajta*/
        {
            if((broj>>(BITOVA-i)) & 1) /*ispitivanje bitova od najviseg*/
                printf("1");
            else
                printf("0");
            if(i%4==0) /*razmak izmedju grupe od 4 bita*/
                printf(" ");
        }
        printf("\n\n");
    }
}
```

Задатак 1_23.

Написати програм на језику C који:

- чита са тастатуре цео број x , величине четири бајта, задат у хексадецималном облику, као и цео број n задат у децималном облику,
- формира број y тако што за прва четири бита у задатом броју x (од бита најмање тежине) мења вредности, а остале битове поставља на нулу,
- формира број z тако што за првих n битова у задатом броју x (од бита најмање тежине) мења вредности, а остале битове поставља на нулу,
- приказује на екрану, у бинарном облику, бројеве x , y и z тако што у сваком од ових бројева одваја сваки бајт (број $0x\ aaaa\ aaaa$ приказује као: 10101010 10101010 10101010 10101010).

Решење 1_23.

```
#include <stdio.h>
#define MASKA1 15 /*vrednost 1 za prva cetiri bita, od bita najmanje tezine*/
#define BITOVA 32 /*opseg od cetiri bajta za podatak*/

main()
{
    int i, iter, n;
    long broj, maska2=0, x, y, z;

    printf("x (u heksadecimalnom obliku): ");
    scanf("%lx", &x);
    do
    {
        printf("n (u decimalnom obliku): ");
        scanf("%d", &n);
    }while(n<1 || n>BITOVA);

    for(i=0; i<n;i++)
        maska2 = maska2 | (1<<i);

    y = (x ^ MASKA1) & MASKA1; /*y ima promenjenu vrednost za prva 4 bita*/
    z = (x ^ maska2) & maska2; /*z ima promenjenu vrednost za prvih n bitova*/

    /*prikaz sva tri broja u binarnom obliku*/
    for(iter=1; iter<=3; iter++) /*u svakoj iteraciji*/
    {
        broj = (iter==1)? x : ((iter==2)? y : z); /*broj dobija vrednost x,y ili z*/

        for(i=1; i<=BITOVA; i++) /*u opsegu od 4 bajta*/
        {
            if((broj>>(BITOVA-i)) & 1) /*ispitivanje bitova od najviseg*/
                printf("1");
            else
                printf("0");
            if(i%8==0) /*razmak izmedju grupa od po 8 bitova*/
                printf(" ");
        }
        printf("\n\n");
    }
}
```

Задатак 1_24.

Написати програм на језику C који:

- чита са тастатуре 16 целих бројева и приказује на екрану (у хексадецималном облику) цео број x , формиран на следећи начин: ако је k -ти задати број у задатом опсегу од n до m , k -ти бит (од бита најмање тежине) у броју x има вредност 1, у супротном овај бит има вредност 0,
- заменаује места нижем и вишем бајту у податку x и приказује на екрану резултат ове замене (у хексадецималном облику),
- понавља кораке под а) и б) све док су вредности n и m у релацији $n < m$.

Решење 1_24.

```
#include <stdio.h>
#define MASKA 0x00ff    /*u binarnom obliku sve jedinice*/
#define VEL_BAJTA 8     /*broj bitova u bajtu*/

main()
{
    int i, n, m, broj, x, pom;

    while(1)
    {
        x = 0;
        printf("\nZadati granice opsega (n,m): ");
        scanf("%d%d", &n, &m);
        if(n>=m)
            break;

        /*Formiranje broja*/
        printf("\nZadati 16 celih brojeva: ");
        for(i=0; i<VEL_BAJTA*2; i++)
        {
            scanf("%d", &broj);
            if(broj>=n && broj<=m)
                x |= (1<<i);
        }

        /*Prikaz originalnog broja x*/
        printf("\n\nx zadat preko tastature: ");
        printf("%#x", x);

        /*Prikaz broja x sa zamenjenim bajtovima*/
        pom = (x & MASKA) << VEL_BAJTA;
        x >>= VEL_BAJTA;
        x |= pom;
        printf("\nx sa zamenjenim bajtovima: %#x", x);
    }
}
```


Задатак 1_25.

Написати програм на језику C који:

- а) чита са тастатуре (у хексадецималном облику) два цела броја x и y , сваки величине два бајта,
- б) чита са тастатуре цео број k и прва 4 бита од позиције k -тог бита у податку x поставља на вредности најнижа 4 бита од податка y . Остатак броја x треба да остане неизмењен;
- в) на основу изабраног једног од три симбола $\&$, $|$, или $\wedge$ приказује на екрану (у хексадецималном облику) резултат примене изабраног оператора, над претходно измењеним бројем x и задатим бројем y .

Решење 1_25.

```
#include <stdio.h>
#define MASKA 0x0f      /*u binarnom obliku: jedinice na mestima prva 4 bita*/

main()
{
    int x, y, poz, op;

    /*Zadavanje potrebnih vrednosti*/
    printf("\nx, y (u hekso obliku)? ");
    scanf("%x%x", &x, &y);

    printf("\nPozicija (u decimalnom obliku)? ");
    scanf("%d", &poz);

    /*Formiranje nove vrednosti x*/
    x |= (y & MASKA) << (poz-1);    /*za redni broj (poz) pomeraj je (poz-1)*/
    printf("\n%x", x);

    /*Prikaz rezultata primene logickog operatora*/
    printf("\n\nZadati operator (&, | ili ^): "); getchar(); op = getchar();

    switch(op)
    {
        case '&':    printf("\nx&y:\t%x\n", x&y); break;
        case '|':    printf("\nx|y:\t%x\n", x|y); break;
        case '^':    printf("\nx^y:\t%x\n", x^y); break;
        default:     printf("Nepoznat operator!"); break;
    }
}
```

Задатак 1_26.

Написати програм на језику C који:

- чита са тастатуре (у хексадецималном облику) два цела броја x и y , сваки величине два бајта,
- мења вредности на местима оних бита у броју x на којима број y има вредност 0 и приказује на екрану резултат ове измене,
- на основу изабраног једног од симбола $\&$, $|$, или $\wedge$ приказује на екрану (у хексадецималном облику) резултат примене изабраног оператора над претходно формираним бројем x и задатим бројем y .

Решење 1_26.

```
#include <stdio.h>
#define MASKA 0x0f      /*u binarnom obliku: jedinice na mestima prva 4 bita*/

main()
{
    int x, y, op;

    /*Zadavanje potrebnih vrednosti*/
    printf("\nx, y (u hekso obliku)? ");
    scanf("%x%x", &x, &y);

    /*Formiranje nove vrednosti x*/
    x ^= ~y;
    printf("\n%x", x);

    /*Prikaz rezultata primene logickog operatora*/
    printf("\n\nZadati operator (&, | ili ^): "); getchar(); op = getchar();

    switch(op)
    {
        case '&':    printf("\nx&y:\t%x", x&y); break;
        case '|':    printf("\nx|y:\t%x", x|y); break;
        case '^':    printf("\nx^y:\t%x", x^y); break;
        default:     printf("Nepoznat operator!"); break;
    }
}
```

Задатак 1_27.

Написати програм на језику C који:

- чита са тастатуре вредности n , x_{min} , x_{max} и dx ,
- табелира на екрану функцију $f(x, n) = \sum_{i=1}^n \frac{x^i}{(2i-1) \cdot 4^{2i-1}}$,

за све вредности $x_{min} \leq x \leq x_{max}$, са кораком dx ,

- понавља све кораке док се вредност n задаје у опсегу од 1 до 100.

Решење 1_27.

```
#include <stdio.h>
#include <math.h>
#define MAX 100 /*dozvoljen opseg broja*/

main()
{
    double xmin,xmax,x,y,dx;
    int i,n;

    while(1)
    {
        printf("\nUnesite n: ");
        scanf("%d",&n);

        if(n<1 || n>MAX)
            break;

        printf("Unesite granice intervala [xmin, xmax] i korak dx: ");
        scanf("%lf%lf%lf",&xmin,&xmax,&dx);

        printf("\n****x****t****y****\n");
        for(x=xmin,y=0;x<=xmax;x+=dx)
        {
            for(i=1;i<=n;i++)
                y+=pow(x,i)/((2*i-1)*pow(4,2*i-1));

            printf("%8.5lf\t%8.5lf\n",x,y);
        }
    }
}
```

Задатак 1_28.

Написати програм на језику C који:

- чита са тастатуре прво цео број n ($n \leq 10$) а онда и n реалних бројева,
- одређује и приказује на екрану колико реалних бројева претходе првом негативном (ако су сви задати бројеви ненегативни онда је то број свих задатих).

Решење 1_28.

```
#include <stdio.h>
#define MAX 10 /*dozvoljeno brojeva*/

main()
{
    int i, n, prvi_neg=0, broj_preth_neg=0;
    float x;
```

```

do
{
    printf("\nKoliko ce biti brojeva? ");
    scanf("%d", &n);
}while(n<1 || n>MAX);

printf("\nZadati brojeve");
for(i=1; i<=n; i++)
{
    printf("\n%d. broj: ", i);
    scanf("%f", &x);

    if(x>=0 && !prvi_neg)
        broj_preth_neg++;
    else
        prvi_neg=1;
}
printf("\nPrvom negativnom prethodi %d brojeva\n", broj_preth_neg);
}

```

Задатак 1_29.

Написати програм на језику C који:

- чита са тастатуре граничне целобројне вредности опсега ($a < 0$ и $b < 0$), а онда и задате целе бројеве, до броја 0 који је стоп за унос,
- одређује суму свих задатих негативних бројева чије су вредности у опсегу од a до b ,
- одређује вредност највећег негативног броја од свих задатих.

Решење 1_29.

```

#include <stdio.h>

main()
{
    int a, b, x, s=0, broj_neg=0, max;

    do
    {
        printf("\nZadati opseg (a<0,b<0,a<b)? ");
        scanf("%d%d", &a, &b);
    }while(a>=0 || b>=0 || a>=b);

    while(1)
    {
        printf("\nBroj? ");
        scanf("%d", &x);
        if(x==0)
            break;
        if(x<0 && x>=a && x<=b)
        {
            s+=x;
            if(broj_neg==0) /*ako nije bilo negativnih*/
                max = x; /*prvi zadati se proglasava najvecim*/
        }
    }
}

```

```

        else if(x>max)      /*ako je nadjen veci od max*/
            max = x;        /*onda je to novi max*/
        broj_neg++;
    }
}
printf("\nNegativnih brojeva iz zadatog opsega: %d", broj_neg);
printf("\nSuma negativnih brojeva iz zadatog opsega: %d", s);
printf("\nNajveci negativni broj iz zadatog opsega: %d", max);
}

```

Задатак 1_30.

Написати програм на језику C који:

- чита са тастатуре прво цео број n ($n \leq 10$) а онда и n целих бројева,
- одређује и приказује на екрану бројеве ($n0$, $n1$ и $n2$) оних задатих бројева који при дељењу бројем 3 дају остатак 0, 1 и 2, респективно,
- програм понавља кораке под а) и б) све док је број n у траженом опсегу.

Решење 1_30.

```

#include <stdio.h>
#define MAX 10          /*dozvoljeno brojeva*/

main()
{
    int i, n, broj, n0, n1, n2;

    while(1)
    {
        printf("\n\nKoliko ce biti brojeva? ");
        scanf("%d", &n);
        if(n<1 || n>MAX)
            break;

        n0=n1=n2=0;          /*inicijalizacija u svakoj iteraciji*/
        for(i=0; i<n; i++)    /*citanje n brojeva*/
        {
            printf("%d. broj: ", i+1);
            scanf("%d", &broj);

            /*odredjivanje ostatka deljenja brojem 3*/
            if(broj%3==0)
                n0++;
            else if(broj%3==1)
                n1++;
            else
                n2++;
        }
        printf("\nn0: %d\tn1: %d\tn2: %d", n0, n1, n2);
    }
}

```

Задатак 1_31.

Написати програм на језику С који:

- а) чита са тастатуре прво цео број n ($n \leq 50$) а онда и n реалних бројева,
- б) одређује и приказује на екрану бројеве (nm , nv и nj) парова суседних међу задатим бројевима, код којих важи релација $<$, $>$ и $=$,
- в) програм понавља кораке под а) и б) све док се не задају сви реални бројеви међусобно еквивалентних вредности.

Решење 1_31.

```
#include <stdio.h>
#define MAX 50 /*dozvoljeno brojeva*/

main()
{
    int i, n, nm, nv, nj;
    float a, b;

    while(1)
    {
        do
        {
            printf("\n\nKoliko ce biti brojeva? ");
            scanf("%d", &n);
        }while(n<1 || n>MAX);

        printf("1. broj: ");
        nm=nv=nj=0; /*inicijalizacija u svakoj iteraciji*/
        scanf("%f", &a); /*citanje prvog realnog broja*/

        for(i=1; i<n; i++) /*citanje ostalih realnih brojeva*/
        {
            printf("%d. broj: ", i+1);
            scanf("%f", &b);

            /*poredjenje prethodno ucitanog i tekuceg*/
            if(a<b)
                nm++;
            else if(a>b)
                nv++;
            else
                nj++;
            a = b; /*tekuci ce biti prethodni u sledecoj iteraciji*/
        }

        if(nj==n-1) /*svaki par je od medjusobno jednakih*/
            break;

        printf("Parova susednih brojeva kod kojih je");
        printf("\n  jedan broj manji od svog sledbenika: %d", nm);
        printf("\n  jedan broj veci od svog sledbenika: %d", nv);
        printf("\n  jedan broj jednak svom sledbeniku: %d", nj);}
}
```

Задатак 1_32.

Написати програм на језику C који:

- чита са тастатуре текст до знака тачка (.) као ознаке за крај, одређује и приказује на екрану број децималних цифара у задатом тексту, као и средњу вредност свих ових цифара,
- исписује на екрану поруку о томе да ли текст међу узастопним цифрама има међусобно једнаких или не.

Решење 1_32.

```
#include <stdio.h>
#include <ctype.h>

main()
{
    int c, preth_c='a', n=0, jednakih=0;
    float suma=0;

    printf("Zadati tekst:\n");
    while((c=getchar())!='.') /*citanje znak po znak do tacke*/
    {
        if(isdigit(c)) /*ako je znak decimalna cifra*/
        {
            n++;
            suma += c - '0'; /*izracunavanje vrednosti procitane cifre*/

            if(c==preth_c) /*ako su uzastopni medjusobno jednaki*/
                jednakih=1;
            preth_c=c;
        }
    }
    printf("\nBroj decimalnih cifara do tacke je: %d",n)
    printf("\nSrednja vrednost je: %f",suma/n);

    if(jednakih)
        printf("\nIma medjusobno jednakih uzastopnih cifara.\n");
    else
        printf("\nNema medjusobno jednakih uzastopnih cifara.\n");
}
```

Задатак 1_33.

Написати програм на језику C који:

- чита са тастатуре један текст до знака за прелаз у нови ред ($\backslash n$) као ознаке за крај и копира овај текст на екран, при чему сваку децималну цифру конвертује у одговарајући број празних знакова (на пример: цифру 5 конвертује у 5 празних знакова),
- приказује на екрану бројеве цифара 1, 2, 3, као и свих осталих децималних цифара заједно, у задатом тексту.

Решење 1_33.

```
#include <stdio.h>
#include <ctype.h>

main()
{
    int i, c, n1=0, n2=0, n3=0, nost=0, praznih;

    printf("Zadati tekst:\n");
    while((c=getchar())!='\n')           /*citanje znak po znak do tacke*/
    {
        if(isdigit(c))                  /*ako je znak decimalna cifra*/
        {
            praznih = c - '0';          /*izracunavanje njene vrednosti*/

            for(i=0; i<praznih; i++)     /*odgovarajuci broj praznih mesta*/
                putchar(' ');

            /*odredjivanje trazenih brojeva cifara*/
            if(c=='1')
                n1++;
            else if(c=='2')
                n2++;
            else if(c=='3')
                n3++;
            else
                nost++;
        }
        else
            putchar(c);
    }

    printf("\nBroj cifara 1 je: %d", n1);
    printf("\nBroj cifara 2 je: %d", n2);
    printf("\nBroj cifara 3 je: %d", n3);
    printf("\nBroj ostalih cifara je: %d", nost);
}
```

Задатак 1_34.

Написати програм на језику C који:

- чита са тастатуре ред знакова (до знака за прелаз у нови ред '\n' као ознаке за крај) и приказује на екрану број децималних цифара у овом реду,
- приказује на екрану број формиран од првих n ($n \leq 10$) децималних цифара које садржи тај ред знакова.

Решење 1_34.

```
#include <stdio.h>
#include <ctype.h>
#define MAX 10                                /*dozvoljen broj cifara */

main()
{
    int c, cifra, n=0, broj=0;

    printf("Zadati tekst:\n");
    while((c=getchar())!='\n')                /*citanje znak po znak do tacke*/
    {
        if(isdigit(c))                        /*ako je znak decimalna cifra*/
        {
            n++;
            if(n>10)
                break;
            cifra = c - '0';                  /*izracunavanje vrednosti svake cifre*/
            broj = broj*10 + cifra;           /*formiranje broja od svih cifara redom*/
        }
    }
    printf("\nBroj decimalnih cifara:  %d", n);
    printf("\nBroj formiran od ovih cifara: %d\n", broj);
}
```


2. Нумерички и знаковни низови, сортирање и претраживање низова у програмима на језику С

Задатак 2_1.

Написати програм на језику C који:

- а) чита са тастатуре два цела броја x и y и приказује на екрану вредности бројева z и w , који су добијени као резултати целобројног и реалног дељења броја x бројем y ,
- б) понавља корак под а) десет пута или до задате вредности 0 за број y ,
- в) приказује на екрану низ формиран од остатака дељења броја x бројем y , у свим итерацијама, редом.

Решење 2_1.

```
#include<stdio.h>
#define MAX 10                /*dozvoljen broj deljenja*/

main()
{
    int x, y, z, i, j, ostaci[MAX];
    float w;

    for(i=0;i<MAX;i++)
    {
        printf("\n\nUnesite dva broja: ");
        scanf("%i%i",&x,&y);

        if(y==0)
        {
            printf("\nDeljenje nulom nije dozvoljeno\n\n");
            break;
        }

        z=x/y;                /*celobrojni deo kolicnika*/
        w=(float)x/y;          /*realni deo kolicnika*/

        printf("\nCelobrojno %i podeljeno sa %i je %i",x,y,z);
        printf("\nRealno %i podeljeno sa %i je %f",x,y,w);

        ostaci[i]=x%y;         /*(i+1)-i element niza ostaci*/
    }

    /*stampanje rezultata*/
    if(i!=0)
    {
        printf("\n\nNiz ostataka deljenja je:\n");
        for(j=0;j<i;j++)
            printf("%d ",ostaci[j]);
        printf("\n\n");
    }
}
```

Задатак 2_2.

Написати програм на језику C који:

- приказује на екрану број y добијен инвертовањем цифара од позитивног целог броја x , који се задаје преко тастатуре и састоји од највише 4 цифре (за $x = 123$, $y = 321$),
- приказује на екрану низ z , формиран од оних цифара задатог броја x (од цифре на месту јединице) које су различите од вредности 0,
- понавља кораке под а) и б) све док је задати број y траженом опсегу од 0 до n (задатом преко тастатуре).

Решење 2_2.

```
#include <stdio.h>
#define MAX 20 /*dozvoljen broj cifara je 20*/

main()
{
    int i=0, j, n, x, pom, y=0, z[MAX];

    printf("Unesite opseg n: ");
    scanf("%d",& n);

    while(1)
    {
        printf("\n\nUnesite jedan ceo broj: ");
        scanf("%d",&x);

        if(x < 1 || x > n)
            break;

        pom=x; /*pomocna promenljiva cuva vrednost x*/
        while(pom!=0) /*sve dok ima cifara*/
        {
            y=y*10+pom%10;

            if(pom%10 != 0) /*ako cifra na mestu jedinice nije 0*/
                z[i++]=pom%10; /*ova cifra postaje element niza z*/

            pom=pom/10; /*pomeranje cifara broja udesno*/
        }

        /*stampanje rezultata*/
        printf("\nBroj y: ");
        printf("y= %d\n",y);

        printf("Niz z: ");
        for(j=0;j<i;j++)
            printf("%d ",z[j]);
    }
}
```

Задатак 2_3.

Написати програм на језику C који:

- а) чита са тастатуре цео број x и приказује на екрану број y формиран уклањањем свих нула са десне стране броја x ($x = 500$, $y = 5$),
- б) приказује на екрану вредности елемената низа z , формираног од бројева појављивања појединих децималних цифара у броју y ,
- в) понавља кораке под а) и б) до задате непозитивне вредности за број x .

Решење 2_3.

```
#include <stdio.h>

main()
{
    int x, y, pom, i, z[10], br_nizova=0;

    while(1)
    {
        printf("\nUnesite jedan ceo broj: ");
        scanf("%d",&x);

        if (x<=0)
            break;                /*kraj petlje kada se ucita broj ne veci od 0*/

        pom=y=x;                  /*pomocna promenljiva cuva vrednost x*/
        while(pom%10==0)          /*ako je cifra na mestu jedinice 0*/
        {
            y=pom/10;             /*deljenje brojem 0 uklanja ovu cifru*/
            pom/=10;               /*pomeranje cifara broja udesno*/
        }

        printf ("Broj %d bez nula sa desne strane %d\n",x,y);

        pom=y;                    /*pomocna promenljiva cuva vrednost y*/
        for (i=0;i<10;i++)
            z[i]=0;                /*broj svake cifre na pocetku je 0*/

        while (pom)
        {
            z[pom%10]++;          /*sama cifra je indeks jednog elementa z*/
            pom=pom/10;           /*pomeranje cifara broja udesno*/
        }

        /*stampanje rezultata*/
        printf("\nNiz z:\n");
        for(i=0;i<10;i++)
            printf("Cifara %d ima:\t%d\n", i, z[i]);
    }
}
```

Задатак 2_4.

Написати програм на језику C који:

- а) чита са тастатуре један позитиван цео број,
- б) одређује и приказује на екрану децималну цифру која је највише заступљена и ону која је најмање заступљена у задатом броју,
- в) понавља кораке под а) и б) док се учитава број у траженом опсегу (задатом преко тастатуре).

Решење 2_4.

```
#include <stdio.h>

main()
{
    int i, n, broj, cifre[10], max, min;

    printf("Unesite opseg za broj: ");
    scanf("%d",&n);

    for(i=0;i<10;i++)
        cifre[i]=0;           /*broj svake cifre na pocetku je 0*/

    while(1)
    {
        printf("Unesite jedan ceo broj: ");
        scanf("%d",&broj);

        if(broj<1 || broj>n)    /*ako je zadat broj van opsega*/
            break;            /*kraj petlje*/

        while(broj>0)          /*sve dok ima cifara*/
        {
            max=min=broj%10; /*pretpostavljeni max i min*/
            cifre[broj%10]++; /*sama cifra je indeks jednog elementa z*/
            broj/=10;         /*pomeranje cifara broja udesno*/
        }

        for(i=0;i<10;i++)
        {
            if(cifre[i]>cifre[max])
                max=i;
            if(cifre[i]!=0 && cifre[i]<cifre[min])
                min=i;
        }

        /*stampava rezultata*/
        printf("Najmanje puta se pojavila cifra %d\n", min);
        printf("Najvise puta se pojavila cifra %d\n", max);
    }
}
```

Задатак 2_5.

Написати програм на језику C који:

- од n ($n \leq 10$) целих бројева, задатих преко тастатуре, прихвата у низ x само оне бројеве чије су вредности у опсегу од 0 до 100,
- одређује и приказује на екрану број елемената задатог низа x чија је вредност једнака највећој у истом низу,
- формира низ y од индекса оних елемената низа x чија је вредност једнака највећој у низу x и приказује на екрану хистограм вредности свих елемената низа y помоћу знака $*$.

Решење 2_5.

```
#include <stdio.h>
#define MAX 10                                /*dozvoljen broj elemenata niza*/

main()
{
    int i, j=0, k=0, n, broj;
    int x[MAX], y[MAX], max, broj_max=0;

    do
    {
        printf("\nBroj elemenata niza? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX);

    /*citanje niza i odredjivanje najvece vrednosti medju njegovim elementima*/
    printf("\nElementi niza? ");
    for(i=0; i<n; i++)
    {
        scanf("%d", &broj);

        if(broj>=0 && broj<=100)
        {
            x[j] = broj;
            if(j==0)
                max=x[j];
            else if(x[j]>max)
                max = x[j];
            j++;
        }
    }

    /*formiranje niza od indeksa svih elemenata sa najvecom vrednoscu*/
    for(i=0; i<j; i++)
        if(x[i]==max)
        {
            broj_max++;
            y[k++] = i;
        }

    /*stampanje rezultata*/
    printf("Niz x:\n");
    for(i=0; i<j; i++)
        printf("\nx[%d]: %d", i, x[i]);
```



```

printf("\n\nNajvecu vrednost ima %d elemenata.\n\n", broj_max);

printf("Histogram vrednosti elemenata niza y:");
for(i=0; i<k; i++)
{
    printf("\ny[%d] (%d):\t", i, y[i]);

    for(j=0; j<y[i]; j++)
        printf("*");
}
printf("\n\n");
}

```

Задатак 2_6.

Написати програм на језику C који:

- од n ($n \leq 50$) реалних бројева, задатих преко тастатуре, прихвата у низ x само бројеве чије су вредности ван опсега од 0 до 50,
- одређује и приказује на екрану средњу вредност задатог низа x , као и број његових елемената, чија је вредност мања од добијене средње вредности,
- формира низ y од редних бројева оних елемената низа x чија је вредност мања од средње вредности низа x и приказује на екрану низ y на следећи начин: највише по 10 вредности у једном реду.

Решење 2_6.

```

#include <stdio.h>
#define MAX 50 /*dozvoljen broj elemenata niza*/

main()
{
    int i, j=0, k=0, n, broj_manjih=0, y[MAX];
    float broj, x[MAX], srednja=0;

    do
    {
        printf("\nBroj elemenata niza? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX);

    /*citanje niza i odredjivanje srednje vrednosti njegovih elemenata*/
    printf("\nElementi niza? ");
    for(i=0; i<n; i++)
    {
        scanf("%f", &broj);

        if(broj<0 || broj>MAX)
        {
            x[j++]=broj;
            srednja += broj;
        }
    }
}

```

```

    if(j!=0)
    {
        srednja /= j;

        /*Niz od rednih brojeva elemenata manje vrednosti od srednje*/
        for(i=0; i<j; i++)
            if(x[i]<srednja)
            {
                broj_manjih++;
                y[k++] = i+1;
            }

        /*stampanje rezultata*/
        printf("\nSrednja vrednost zadatog niza: %f\n", srednja);
        printf("\nElementa manjih od srednje vrednosti: %d\n", broj_manjih);

        printf("\nNiz y:\n");
        for(i=0; i<k; i++)
        {
            printf("%d ",y[i]);

            if(i%10==9) /*stampanje po 10 vrednosti u jednom redu*/
                printf("\n");
        }
        printf("\n\n");
    }
    else
        printf("\nZadati niz nema elemenata trazених vrednosti\n");
}

```

Задатак 2_7.

Написати програм на језику C који:

- чита са тастатуре n ($n \leq 10$) целих бројева и смешта их у низ x ,
- одређује и приказује на екрану индекс највећег и индекс најмањег од свих ненегативних елемената низа x (ако има више елемената са највећом или најмањом вредношћу узима се у обзир елемент са мањим редним бројем),
- понавља кораке под а) и б) m ($m \leq 5$) пута и од нађених индекса највећих и индекса најмањих елемената у m задатих низова редом формира и приказује на екрану низове y и z .

Решење 2_7.

```

#include <stdio.h>
#define MAX1 10          /*dozvoljen broj elemenata niza*/
#define MAX2 5          /*dozvoljen broj nizova*/

main()
{
    int i, j, k=0, l=0, n, imin, imax;
    int x[MAX1], y[MAX2], z[MAX2];

```

```

for(i=0; i<MAX2; i++)
{
    do
    {
        printf("\nBroj elemenata niza? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX1);

    printf("\nElementi niza: ");
    for(j=0; j<n; j++)
        scanf("%d", &x[j]);

    /*Pretpostavka: najvecu i najmanju vrednost ima 1. element*/
    imin=imax=0;
    for(j=1; j<n; j++)
        if(x[j]>=0 && x[j]>x[imax]) /*ako je pronadjena veca vrednost*/
            imax = j;             /*element se proglašava najvećim*/
    for(j=1; j<n; j++)
        if(x[j]>=0 && x[j]<x[imin]) /*ako je pronadjena manja vrednost*/
            imin = j;             /*element se proglašava najmanjim*/

    /*formiranje elemenata nizova u (i+1)-oj iteraciji spoljne petlje*/
    y[k++] = imax;
    z[l++] = imin;
}
/*stampanje rezultata*/
printf("\nNiz y: ");
for(i=0; i<k; i++)
    printf("%d ", y[i]);
printf("\nNiz z: ");
for(i=0; i<l; i++)
    printf("%d ", z[i]);
}

```

Задатак 2_8.

Написати програм на језику C који:

- чита са тастатуре m nizova целих бројева, сваки дужине n ($m \leq n \leq 20$),
- формира низ z од средњих вредности задатих m nizova редом и приказује на екрану по 5 елемената низа у једном реду,
- понавља кораке под а) и б) све док су вредности свих елемената низа z у опсегу задатом преко тастатуре ($0 \dots k$).

Решење 2_8.

```

#include <stdio.h>
#define MAX 20                /*dozvoljena duzina niza*/

main()
{
    int i,j=0,m,n,k,niz[MAX],van_opsega=0;

```

```

float z[MAX],srednja;

while(1)
{
    printf("Unesite vrednosti za m i n (m<=n<=20): ");
    scanf("%d%d",&m,&n);

    if (m<1 || m>=n || n>20)
        break;

    printf("Unesite k za opseg (k>0): ");
    do
        scanf("%d",&k);
    while(k<=0);

    for(i=0;i<m;i++)
    {
        printf("Unesite %d elemenata %d niza:\n",n,i+1);

        srednja=0;
        for (j=0;j<n;j++)
        {
            scanf("%d",&niz[j]);
            srednja+=niz[j];
        }
        z[i]=srednja/n;      /*formiranje niza z od srednje vrednosti*/

        if (z[i]<0 || z[i]>k)
            van_opsega=1;
    }

    if(van_opsega)
    {
        printf("Sve srednje vrednosti nisu u opsegu 0..%d\n",k);
        break;
    }

    /*stampa rezultata*/
    printf("\nNiz od srednjih vrednosti:\n");
    for (i=0;i<m;i++)
    {
        printf("%.2f ",z[i]);
        if(i%10==4)      /* po 5 vrednosti u jednom redu*/
            printf("\n");
    }
    printf("\n");
}
}

```

Задатак 2_9.

Написати програм на језику C који:

- а) чита са тастатуре дужину низа n ($n \leq 100$) као и сам низ x од n целих бројева,
- б) инвертује поредак елемената у низу x (треба да замени места елементима: првом и последњем, другом и претпоследњем...),
- в) понавља кораке под а) и б) m пута и приказује на екрану низ у формиран од инвертованих низова x , у свим итерацијама редом (низови додати један на крај другог).

Решење 2_9.

```
#include <stdio.h>
#define MAX 10                                /*dozvoljena duzina niza, kao i broj nizova*/

main()
{
    int i, j, k=0, n, x[MAX], y[MAX*MAX], pom;

    for(i=0; i<MAX;i++)
    {
        printf("\nDuzina niza (n<=%d)? ", MAX);
        scanf("%d", &n);

        if(n<1 || n>MAX)
            break;

        printf("\nElementi niza: ");
        for(j=0; j<n; j++)
            scanf("%d", &x[j]);

        /*invertovanje mesta elemenata u nizu*/
        for(j=0;j<n/2;j++)
        {
            pom = x[j];
            x[j] = x[n-j-1];
            x[n-j-1] = pom;
        }

        printf("\n%d. invertovani niz: ", k+1);
        for(j=0; j<n; j++)
            printf("%d ", x[j]);

        /*dodavanje elemenata glavnom nizu*/
        for(i=0;i<n;i++)
            y[k++]=x[i];
    }
    printf("\nRezultujuci niz:\n");
    for(i=0; i<k; i++)
        printf("%d ", y[i]);
}
```

Задатак 2_10.

Написати програм на језику C који:

- а) чита са тастатуре n ($n \leq 10$) реалних бројева и смешта их у низ x ,
- б) даје извештај на екрану о томе да ли међу елементима задатог низа x има међусобно једнаких суседних вредности,
- в) формира и приказује на екрану низ y од индекса оних елемената задатог низа x од којих сваки има исту вредност као његов претходни елемент, као и низ z од индекса осталих елемената у истом низу.

Решење 2_10.

```
#include <stdio.h>
#define MAX 10 /*dozvoljen broj elemenata niza*/

main()
{
    int i, j=0, k=1, n, y[MAX], z[MAX];
    double x[MAX];

    do
    {
        printf("\nBroj elemenata niza? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX);

    printf("\nElementi niza: ");
    for(i=0; i<n; i++)
        scanf("%lf", &x[i]);

    /*formiranje nizova y i z*/
    z[0] = 0; /*prvi element nema prethodnika*/

    for(i=1; i<n; i++)
        if(x[i]==x[i-1])
            y[j++]=i;
        else
            z[k++]=i;

    /*stampanje rezultata*/
    if(j!=0)
        printf("\nNiz ima medjusobno iste susedne elemente\n");
    else
        printf("\nNiz nema medjusobno iste susedne elemente\n");

    printf("\nNiz y: ");
    for(i=0; i<j; i++)
        printf("%d ", y[i]);
    printf("\nNiz z: ");
    for(i=0; i<k; i++)
        printf("%d ", z[i]);
}
```

Задатак 2_11.

Написати програм на језику C који:

- а) чита са тастатуре низ x од n ($n \leq 10$) целих бројева, као и цео број k ,
- б) приказује на екрану број међусобно различитих елемената низа x ,
- в) формира низ y од низа x тако што после сваког елемента низа x који је дељив бројем k убацује вредност 1, а после сваког елемента који није дељив бројем k убацује вредност 0.

Решење 2_11.

```
#include <stdio.h>
#define MAX 10 /*dozvoljen broj elemenata niza*/

main()
{
    int x[MAX], y[MAX], i, j=0, n;
    int k, nadjeni_isti, broj_razl=0;

    /*Citanje duzine niza i celog broja k*/
    do
    {
        printf("\nBroj elemenata niza? ");
        scanf("%d", &n);
    }while(n<1 || n>MAX);

    printf("\nVrednost k? ");
    scanf("%d", &k);

    /*Citanje niza*/
    printf("\nElementi niza? ");
    for(i=0; i<n; i++)
        scanf("%d", &x[i]);

    /*Nalazenje broja razlicitih elemenata niza x*/
    for(i=0; i<n-1; i++)
    {
        nadjeni_isti = 0;
        for(j=i+1; j<n; j++)
        {
            if(x[i]==x[j]) /*ako se vrednost elementa ponavlja dalje u nizu*/
            {
                nadjeni_isti=1;
                break;
            }
        }
        if(!nadjeni_isti) /*ako se vrednost ne ponavlja dalje u nizu*/
            broj_razl++;
    }
    broj_razl++;

    /*Formiranje i prikaz novog niza*/
    for(i=0, j=0; i<n; i++)
    {
        y[j++] = x[i];
```

```

        if(x[i]%k==0)      /*ako je vrednost deljiva brojem k*/
            y[j++]=1;      /*dodavanje jedinice*/
        else               /*ako to nije slucaj*/
            y[j++]=0;      /*dodavanje nule*/
    }

    /*stampanje rezultata*/
    printf("\nElementi zadatog niza: ");
    for(i=0; i<n; i++)
        printf("%d ", x[i]);

    printf("\n\nRazlicitih elemenata niza x: %d\n", broj_razl);

    printf("\nElementi novog niza: ");
    for(i=0; i<j; i++)
        printf("%d ", y[i]);
}

```

Задатак 2_12.

Написати програм на језику C који:

- чита са тастатуре два низа: низ x од n ($n \leq 100$) целих бројева и низ y од m ($m \leq 10$) целих бројева, као и цео број k ($k \leq n$),
- формира низ z од елемената низа x , у који су од k -тог елемента убачени сви елементи низа y ,
- сортира елементе низа z у неоппадајући поредак и приказује резултујући низ на екрану.

Решење 2_12.

```

#include <stdio.h>
#define MAX1 100          /*dozvoljen broj elemenata jednog niza*/
#define MAX2 10           /*dozvoljen broj elemenata drugog niza*/

main()
{
    int x[MAX1], y[MAX2], z[MAX1+MAX2];
    int i, j=0, k, n, m, pom;

    do
    {
        printf("\nDuzine nizova? ");
        scanf("%d%d", &n, &m);
    }while(n<1 || n>MAX1 || m<1 || m>MAX2);

    do
    {
        printf("\nVrednost k? ");
        scanf("%d", &k);
    }while(k>n);
}

```



```

/*Citanje 1. niza*/
printf("\nNiz od %d brojeva?\n", n);
for(i=0; i<n; i++)
    scanf("%d", &x[i]);
/*Citanje 2. niza*/
printf("\nNiz od %d brojeva?\n", m);
for(i=0; i<m; i++)
    scanf("%d", &y[i]);

/*Formiranje novog niza*/
for(i=0; i<k-1; i++)
    z[j++]=x[i];
for(i=0; i<m; i++)
    z[j++]=y[i];
for(i=k-1; i<n; i++)
    z[j++]=x[i];

/*Stampanje elemenata novog niza*/
printf("\nElementi novog niza:\n");
for(i=0; i<j; i++)
    printf("%d ", z[i]);
n=j;

/*Sortiranje novog niza u neopadajuci poredak*/
for(i=0; i<n-1; i++)
{
    for(j=i+1; j<n; j++)
    {
        if(z[i]>z[j])           /*ako nije trazeni poredak elemenata*/
        {
            pom = z[i];      /*zamena mesta elemenata*/
            z[i] = z[j];
            z[j] = pom;
        }
    }
}
/*Stampanje sortiranog niza*/
printf("\n\nSortirani elementi novog niza:\n");
for(i=0; i<j; i++)
    printf("%d ", z[i]);
}

```

Задатак 2_13.

Написати програм на језику C који:

- чита са тастатуре n ($n \leq 50$) целих бројева и смешта их у низ x ,
- формира и приказује на екрану низ у од следећих елемената задатог низа x : од првог појављивања елемента најмање вредности до последњег појављивања елемента највеће вредности,
- сортира низ у нерастући поредак и приказује на екрану резултујући низ.

Решење 2_13.

```
#include <stdio.h>
#define MAX 50 /*dozvoljena duzina niza*/
main()
{
    int i,j, k=0, n, x[MAX], y[MAX], pom, prvi_min, posl_max;

    do
    {
        printf("\nUnesite broj elemenata niza (0<n<=50): ");
        scanf("%d",&n);
    }while(n<1 || n>50);

    printf("\n\nUnesite elemente niza:\n");
    for(i=0;i<n;i++)
        scanf("%d",&x[i]);

    prvi_min=posl_max=0; /*pretpostavljene vrednosti*/
    for(i=0;i<n;i++)
    {
        if(x[i]<x[prvi_min]) /*samo prvo pojavljivanje minimuma*/
            prvi_min=i;
        if(x[i]>=x[posl_max]) /*poslednje pojavljivanje maksimuma*/
            posl_max=i;
    }

    printf("\nmin se prvi put pojavljuje na %d. mestu\n",prvi_min+1);
    printf("\nmax se poslednji put pojavljuje na %d mestu\n",posl_max+1);

    for(i=prvi_min;i<=posl_max;i++)
        y[k++]=x[i];

    /*stampanje rezultata*/
    printf("\nElementi niza y (x od prvog min do poslednjeg max):\n");
    for(i=0;i<k;i++)
        printf("%d ",y[i]);

    /*dovodjenje elemenata u nerastuci poredak*/
    for(i=0;i<k-1;i++)
        for(j=i+1;j<k;j++)
            if(y[i]<y[j]) /*ako je nedozvoljena relacija*/
            {
                pom=y[i]; /*zamena mesta elementima u nizu*/
                y[i]=y[j];
                y[j]=pom;
            }

    /*stampanje rezultata*/
    printf("\n\nElementi niza y u nerastucem poretku:\n");
    for(i=0;i<k;i++)
        printf("%d ",y[i]);
    printf("\n\n");
}
```

Задатак 2_14.

Написати програм на језику C који:

- а) чита са тастатуре дужину низа n ($n \leq 100$) као и сам низ x од n целих бројева,
- б) формира низ y од индекса свих елемената низа x , чија је вредност 0,
- в) израчунава суму свих елемената који су између прве и друге нуле у низу x ,
- г) формира низ z од елемената низа x на следећи начин: сваку серију узастопних нула замењује бројем нула у серији.

Решење 2_14.

```
#include <stdio.h>
#define MAX 100          /*dozvoljena duzina niza*/

main()
{
    int i,j=0,k=0,n,x[MAX],y[MAX],z[MAX],nula=0,suma=0;

    do
    {
        printf("Unesite broj elemenata niza (od 1 do 100): ");
        scanf("%d",&n);
    }while(n<1 || n>MAX);

    printf("\nUnesite elemente niza: ");
    for(i=0;i<n;i++)
        scanf("%d",&x[i]);

    /*formiranje niza y*/
    for(i=0;i<n;i++)
        if(x[i]==0)
            y[j++]=i;

    /*stampanje rezultata*/
    printf("\n\n");
    for(i=0;i<j;i++)
        printf("U nizu x nula se %d put pojavila na %d mestu\n",i+1,y[i]+1);

    /*formiranje i prikaz sume*/
    if (j<2)
        printf("\nZadati niz nema dve nule\n");
    else
    {
        for(i=y[0];i<y[1];i++)
            suma+=x[i];
        printf("\nSuma izmedju prve i druge nule: %d\n",suma);
    }

    /*formiranje i prikaz niza z*/
    for(i=0;i<n;i++)
    {
        if(x[i]!=0 && nula==0)      /*za nenula element niza van serije nula*/
            z[k++]=x[i];
    }
```

```

        else
            if(x[i]==0)                /*za nula element*/
                nula++;
            else                        /*za nenula element niza после серије нула*/
            {
                z[k++]=nula;
                z[k++]=x[i];
                nula=0;
            }
        }
    if(x[n-1]==0)                      /*ако се низ завршава нулом*/
        z[k++]=nula;
    printf("\nNiz sa zamenjenom serijom nula brojem njih u seriji:\n");
    for(i=0;i<k;i++)
        printf("%d ",z[i]);
    printf("\n\n");
}

```

Задатак 2_15.

Написати програм на језику C који:

- а) чита са тастатуре низ a од n ($n \leq 15$) елемената који представљају коефицијенте полинома $P(x)$ степена n :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

- б) формира низ b , чији су елементи коефицијенти полинома $P'(x)$ који је извод полинома $P(x)$:

$$P'(x) = b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \dots + b_1 x + b_0.$$

на следећи начин: $b[0]=a[1]$, $b[1]=a[2]*2$, $b[2]=a[3]*3 \dots b[n-2]=a[n-1]*(n-1)$

- в) чита са тастатуре низ c од m ($m \leq 15$) елемената који представљају коефицијенте полинома $Q(x)$ степена m : $Q_m(x) = b_m x^m + b_{m-1} x^{m-1} + \dots + b_1 x + b_0$, израчунава полином $R_{n+m}(x) = P_n(x) * Q_m(x)$ и приказује на екрану полиноме $P(x)$, $P'(x)$, $Q(x)$ и $R(x)$.

Решење 2_15.

```

#include <stdio.h>
#define MAX 15                /*dozvoljen stepen polinoma*/

main()
{
    float a[MAX+1],b[MAX],c[MAX+1], d[MAX*2+2];
    int i, j, n, m;

    /*citanje stepena i koeficijenata polinoma P*/
    do
    {
        printf("\nUnesite stepen polinoma P: ");
        scanf("%d",&n);
    }while(n<1 || n>MAX);
}

```

```

printf("\nUnesite koeficijente polinoma P(x): ");
for(i=0;i<n+1;i++)
    scanf("%f",&a[i]);
printf("\n");

/*formiranje koeficijenata polinoma P'*/
for(i=1;i<n+1;i++)
    b[i-1]=a[i]*i;

/*citanje stepena i koeficijenata polinoma Q*/
do
{
    printf("\nUnesite stepen polinoma Q: ");
    scanf("%d",&m);
}while(m<1 || m>MAX);

printf("Unesite koeficijente polinoma Q(x)\n");
for(i=0;i<m+1;i++)
    scanf("%f",&c[i]);
printf("\n");

/*formiranje koeficijenata polinoma R koji ima m+n+1 koeficijenata*/
for(i=0;i<n+m+2;i++)
    d[i]=0;

for(i=0;i<n+1;i++)
    for(j=0;j<m+1;j++)
        d[i+j]=d[i+j]+a[i]*c[j];

/*stampanje rezultata*/
printf("P(x)=");
for(i=n;i>0;i--)
    printf("%f*x^%d +",a[i],i);
printf(" %f\n\n",a[0]);

printf("P'(x)=");
for(i=n-1;i>0;i--)
    printf("%f*x^%d +",b[i],i);
printf(" %f\n\n",b[0]);

printf("Q(x)=");
for(i=m;i>0;i--)
    printf("%f*x^%d +",c[i],i);
printf(" %f\n\n",c[0]);

printf("R(x)=");
for(i=n+m;i>0;i--)
    printf("%f*x^%d +",d[i],i);
printf(" %f\n\n",d[0]);
}

```

Задатак 2_16.

Написати програм на језику C који:

- чита са тастатуре целобројне елементе матрице M димензија $n \times m$, где су $0 < n \leq 20$ и $0 < m \leq 20$,
- формира и приказује на екрану низ max чији су елементи максимуми колона матрице M ,
- одређује минимални елемент низа max и приказује на екрану колону матрице M чији је он максимални елемент,
- понавља претходне кораке све док се за димензије матрице уносе бројеви из задатих интервала.

Пример: Нека је дата матрица M 3×4 :

$$M = \begin{bmatrix} 1 & 2 & 5 & 9 \\ 7 & 0 & 1 & 3 \\ 4 & 6 & 2 & 8 \end{bmatrix}$$

Максимални елементи колона матрице M су: 7, 6, 5 и 9. Минимални елемент је 5 и елементи колоне чији је максимум 5 (трећа колона) су: 5, 1, 2.

Решење 2_16.

```
#include <stdio.h>
#define MAX 20          /*dozvoljena dimenzija matrice: 20x20*/

main(){
    int M[MAX][MAX], max[MAX];
    int i,j,n,m,maxkol,imin,min;

    while(1)
    {
        printf("\n\nUnesite broj vrsta i broj kolona\n");
        scanf("%d%d",&n,&m);

        if(n<1 || n>MAX || m<1 || m>MAX)
            break;

        printf("Unesite elemente matrice M\n");
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                scanf("%d",&M[i][j]);

        /*formiranje niza max*/
        for(j=0;j<m;j++)
        {
            maxkol=M[0][j];
            for(i=0;i<n;i++)
                if(maxkol<M[i][j])
                    maxkol=M[i][j];
            max[j]=maxkol;
        }
    }
}
```

```

printf("\nNajveci elementi matrice M po kolonama su:\n");
for(i=0;i<m;i++)
    printf("%d\t",max[i]);

/*odredjivanje minimuma niza max i stampa rezultata*/
for(imin=0,min=max[0],i=0;i<m;i++)
    if(min>max[i])
    {
        min=max[i];
        imin=i;
    }

printf("\n\nMinimalni maksimum po kolonama je %d\n",min);
printf("Redni broj kolone kojoj on pripada: %d\n\n", imin+1);
printf("Sama kolona kojoj on pripada:\n");

for(i=0;i<n;i++)
    printf("\n%d",M[i][imin]);
}
}

```

Задатак 2_17.

Написати програм на језику C који:

- чита са тастатуре елементе матрица $M1$ и $M2$ димензија $n \times n$, где је $0 < n \leq 20$,
- формира матрицу M на следећи начин: елементи изнад главне дијагонале једнаки су одговарајућим елементима изнад главне дијагонале матрице $M1$, а елементи испод главне дијагонале једнаки су одговарајућим елементима испод главне дијагонале матрице $M2$, док се на главној дијагонали наизменично налазе максимуми споредних дијагонала $M1$ и $M2$,
- понавља претходне кораке све док се не унесе вредност n ван опсега.

Пример: Нека су дате матрице $M1$ и $M2$ димензија 3×3 :

$$M_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{и} \quad M_2 = \begin{bmatrix} 2 & 2 & 3 \\ 7 & 6 & 5 \\ 8 & 0 & 1 \end{bmatrix}.$$

Елементи споредне дијагонале матрице M_1 су: 3, 5, 7 (највећи је 7), а споредне дијагонале M_2 : 3, 6, 8 (највећи је 8). Тражена матрица M је:

$$M = \begin{bmatrix} 7 & 2 & 3 \\ 7 & 8 & 6 \\ 8 & 0 & 7 \end{bmatrix}$$

Решење 2_17.

```
#include <stdio.h>
#define MAX 20          /*dozvoljena dimenzija matrice: 20x20*/

main()
{
    int M[MAX][MAX], M1[MAX][MAX], M2[MAX][MAX];
    int i,j,n,max1,max2;

    while(1)
    {
        printf("Unesite dimenziju kvadratne matrice n (0<n<20) ");
        scanf("%d",&n);
        if(n<1||n>20)
            break;

        /*citanje elemenata matrica*/
        printf("Unesite elemente matrice M1\n");
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                scanf("%d",&M1[i][j]);
        printf("\nUnesite elemente matrice M2\n");
        for(i=0;i<n;i++)
            for(j=0;j<n;j++){
                scanf("%d",&M2[i][j]);
            }

        /*formiranje delova iznad i ispod glavne dijagonale matrice M*/
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                if(i<j)                    /*iznad dijagonale*/
                    M[i][j]=M1[i][j];
                else                        /*ispod dijagonale*/
                    M[i][j]=M2[i][j];

        /*odredjivanje maximuma sporednih dijagonala od M1 i M2*/
        /*element je na sporednoj dijagonali ako i+j=n-1 */
        max1=M1[0][0];
        max2=M2[0][0];
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            {
                if(M1[i][j]>max1 && i+j==n-1)
                    max1=M1[i][j];
                if(M2[i][j]>max2 && i+j==n-1)
                    max2=M2[i][j];
            }

        /*postavljanje max1 i max2 na glavnu dijagonalu matrice M*/
        for(i=0;i<n;i++)
            if(i%2==0) M[i][i]=max1;
            else M[i][i]=max2;
    }
}
```



```

        /*stampanje rezultata*/
        printf("\nM=\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
                printf("%d ",M[i][j]);
            printf("\n");
        }
    }
}

```

Задатак 2_18.

Написати програм на језику C који:

- чита са тастатуре елементе матрица $M1$ и $M2$ димензија $n \times n$, где је $0 < n \leq 20$,
- израчунава и приказује на екрану матрицу

$$M = M_1^2 + M_1 * M_2 - M_1^T, \text{ где је } M_1^T$$

транспонована матрица матрице $M1$ а $M1 * M2$ производ матрица.

- понавља претходне кораке све док је вредност n унутар дозвољеног опсега.

Решење 2_18.

```

#include <stdio.h>
#define MAX 20                /*dozvoljena dimenzija matrice 20x20*/

main()
{
    int M1[MAX][MAX], M2[MAX][MAX], M[MAX][MAX], POM[MAX][MAX];
    int i,j,k,n;

    while(1)
    {
        printf("Unesite dimenziju kvadratne matrice n (0<n<20) ");
        scanf("%d",&n);
        if(n<1||n>20)
            break;

        /*citanje elemenata matrica M1 i M2*/
        printf("Unesite elemente matrice M1\n");
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                scanf("%d",&M1[i][j]);

        printf("\nUnesite elemente matrice M2\n");
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            {
                scanf("%d",&M2[i][j]);
                M[i][j]=0;
                POM[i][j]=0;
            }
    }
}

```

```

/*izracunavanje elemenata matrica M=M1*M2 i POM=M1^2*/
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        for(k=0;k<n;k++)
        {
            M[i][j]+=M1[i][k]*M2[k][j];
            POM[i][j]+=M1[i][k]*M1[k][j];
        }

/*izracunavanje elemenata matrice M=M1^2+M1*M2-M1T
transponovanje u istoj petlji promenom redosleda indeksa*/
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        M[i][j]+=POM[i][j]-M1[j][i];
printf("\nM=\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
        printf("%d ",M[i][j]);
    printf("\n");
}
}
}

```

Задатак 2_19.

Написати програм на језику C који:

- а) исписује на екрану листу акција које може извршити над квадратном матрицом,
- б) извршава акцију након изабраног њеног редног броја преко тастатуре:
 1. *Unos matrice*
 2. *Prikaz elemenata glavne dijagonale matrice*
 3. *Prikaz elemenata sporedne dijagonale matrice*
 4. *Prikaz kvadrata matrice*
 5. *Prikaz zadate matrice*
 6. *Prikaz transponovane matrice*
 7. *Kraj*
- в) понавља корак све док корисник не изабере опцију број 7 (*Kraj*).

Решење 2_19.

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 20 /*dozvoljena dimenzija matrice 20x20*/

main()
{
    int m[MAX][MAX],c[MAX][MAX];
    int n,i,j,k,spremna=0, opcija, izlaz=0;

```

```

while(1)
{
    printf("\n1. Unos matrice");
    printf("\n2. Prikaz elemenata glavne dijagonale matrice");
    printf("\n3. Prikaz elemenata sporedne dijagonale matrice");
    printf("\n4. Prikaz kvadrata matrice");
    printf("\n5. Prikaz zadate matrice");
    printf("\n6. Prikaz transponovane matrice");
    printf("\n7. Kraj\n\n");

    scanf("%d",&opcija);

    switch(opcija)
    {
        case 1:
            do
            {
                printf("Unesite dimenziju kvadratne matrice\n");
                scanf("%d",&n);
            }while(n<0||n>MAX);

            printf("Unesite elemente matrice:\n");
            for(i=0;i<n;i++)
                for(j=0;j<n;scanf("%d",&m[i][j++]));
            spremna=1;
            break;
        case 2:
            if (!spremna)
            {
                printf("Prvo unesite matricu");
                break;
            }
            printf("Elementi glavne dijagonale su:\n");
            for(i=0;i<n;i++)
                printf("%d\t",m[i][i]);
            break;
        case 3:
            if (!spremna)
            {
                printf("Prvo unesite matricu");
                break;
            }
            printf("Elementi sporedne dijagonale su:\n");
            for(i=0;i<n;i++)
                printf("%d\t",m[i][n-i-1]);
            break;
        case 4:
            for(i=0;i<n;i++)
                for(j=0;j<n;c[i][j++]=0);
            for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                    for(k=0;k<n;k++)
                        c[i][j]+=m[i][k]*m[k][j];
    }
}

```

```

        printf("Kvadrat zadate matrice:\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;printf("%5d",c[i][j++]));
            printf("\n");
        }
        break;
    case 5:
        printf("Zadata matrica:\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;printf("%5d",m[i][j++]));
            printf("\n");
        }
        break;
    case 6:
        for(i=0;i<n;i++)
            for(j=0;j<n;c[i][j++]=m[j][i]);
        printf("Transponovana matrica od zadate:\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;printf("%5d",c[i][j++]));
            printf("\n");
        }
        break;
    default:
        izlaz=1;
        break;
}

if(izlaz)
    break;
printf("\n\n");
}
}

```

Задатак 2_20.

Написати програм на језику C који:

- чита са тастатуре текст (низ знакова) до знака *EOF* или највише *n* (*n=10*) редова текста,
- приказује на екрану задати текст измењен на следећи начин: сваки празан знак мења у знак доња црта (`_`), сваки знак за хоризонталну табулацију мења у два знака доња црта (`__`) и сваку реченицу приказује у посебном реду (ознака за крај реченице је знак тачка (`.`)),
- приказује на екрану хистограм (помоћу знака `*`) бројева појављивања следећих знакова у задатом тексту: празан знак, знак за хоризонталну табулацију и знак за прелаз у нови ред.

Решење 2_20.

```
#include <stdio.h>
#include <string.h>
#define BROJ      10          /*dozvoljen broj redova*/
#define MAX      80          /*broj znakova u jednom redu*/
main()
{
    int i=0, n, c, broj_p=0, broj_t=0, broj_n=0;
    char string[BROJ*MAX];

    /*zamena znakova u ucitanom tekstu*/
    while((c=getchar()) != EOF) /*citanje do znaka EOF*/
    {
        if(c==' ') /*zamena praznog znaka donjom crtom*/
        {
            string[i++] = '_';
            broj_p++;
        }
        else if(c=='\t') /*zamena hor. tabulatora donjim crtama*/
        {
            string[i++] = '_';
            string[i++] = '_';
            broj_t++;
        }
        else if(c=='.') /*posle tacke prelaz u novi red*/
        {
            string[i++] = '.';
            string[i++] = '\n';
            broj_n++;
        }
        else
            string[i++] = c; /*ostali znakovi ostaju isti*/
    }
    n = i; /*formirana duzina niza znakova*/

    /*stampanje rezultata*/
    for(i=0; i<n; i++)
        putchar(string[i]);

    printf("\n\nPraznih:\t");
    for(i=0; i<broj_p; i++)
        putchar('*');
    printf("\nTabulatora:\t");
    for(i=0; i<broj_t; i++)
        putchar('*');
    printf("\nRecenica:\t");
    for(i=0; i<broj_n; i++)
        putchar('*');
    putchar('\n');
}
```

Задатак 2_21.

Написати програм на језику C који:

- а) чита са тастатуре низ речи до речи *"kraj"* (реч је низ знакова између два бела знака),
- б) одређује број речи које почињу и завршавају се истим знаком (asa, сммс...),
- в) одређује дужину најдуже овакве речи.

Решење 2_21.

```
#include<stdio.h>
#include<string.h>
#define BROJ 100          /*dozvoljen broj reci*/
#define MAX      80       /*max broj znakova u reci*/
#define STOP "kraj"      /*rec koja oznacava kraj unosa*/

main()
{
    int i,n,istih=0,l, lmax=0;
    char rec[MAX+1];

    printf ("\nUnesite tekst, za kraj rec \"kraj\":\n");

    for(i=0;i<BROJ;i++)
    {
        scanf("%s",rec);

        if(strcmp(rec,STOP)==0)
            break;

        n++;
        l=strlen(rec);      /*odredjivanje duzine reci*/

        if(rec[0]==rec[l-1]) /*poredjenje 1. i poslednjeg znaka*/
        {
            istih++;

            if(l>lmax)
                lmax=l;
        }
    }

    printf("\n%d reci sa istim pocetkom i krajem",istih);
    printf(" do reci \"kraj\".\n");
    printf("\nNajduza ovakva rec ima %d znakova.\n\n",lmax);

    if(n==0)
        printf("\nNije zadat niz trazениh reci do reci \"kraj\".\n");
}
```

Задатак 2_22.

Написати програм на језику C који:

- а) чита са тастатуре текст (10 редова знакова или до првог празног реда),
- б) одређује и приказује на екрану број редова у задатом тексту, који су дужине мање од 10, као и број осталих редова,
- в) одређује и приказује на екрану број редова унутар малих заграда,
- г) приказује на екрану редове задатог текста без понављања.

Решење 2_22.

```
#include <stdio.h>
#include <string.h>
#define BROJ      10          /*dozvoljen broj redova*/
#define MAX      80          /*broj znakova u redu*/

main()
{
    int i, j, l, n, n1=0, n2=0, pon;
    char red[10][80];

    for(i=0;i<MAX;i++)
    {
        gets(red[i]);          /*citanje reda znakova*/
        if(strlen(red[i])==0)    /*provvara da li je prazan red*/
            break;
    }
    n=i;
    for(i=0;i<n;i++)
    {
        l=strlen(red[i]);       /*izracunavanje duzine reda*/
        if(l<10)
            n1++;
        if(red[i][0]=='(' && red[i][l-1]==')')
            n2++;
    }
    /*stampa rezultata*/
    printf("Broj redova sa manje od 10 znakova je: %d\n",n1);
    printf("Broj redova sa vise od 10 znakova je: %d\n",n-n1);
    printf("Broj redova unutar malih zagrada: %d\n",n2);

    /*prikaz zadatih redova bez ponavljanja*/
    printf("\nRedovi bez ponavljanja:\n");
    for(i=0;i<n-1;i++)
    {
        pon=0;
        for(j=i+1;j<n;j++)
            if(strcmp(red[i],red[j])==0)
                pon++;
        if(!pon)
            puts(red[i]);       /*red se stampa ako nema ponavljanja*/
    }
    puts(red[n-1]);            /*stampanje poslednjeg reda*/
}
```

Задатак 2_23.

Написати програм на језику C који:

- а) чита са тастатуре један ред знакова,
- б) одређује и приказује на екрану број речи, као и број слова у задатом реду текста (реч је низ знакова између два бела знака: ' ' или '\t'),
- в) приказује на екрану задати ред знакова уз претварање почетних слова речи у иста велика слова, а свих осталих слова речи у иста мала слова.

Решење 2_23.

```
#include <stdio.h>
#include <ctype.h>
#define U_RECI 1 /*status: znak je unutar reci*/
#define VAN_RECI 0 /*status: znak je van reci*/

main()
{
    int c, pocetak=1, poz = VAN_RECI, reci=0, slova=0;

    while((c=getchar()) != '\n')
    {
        if(c==' ' || c=='\t')
        {
            if(poz==U_RECI)
            {
                poz = VAN_RECI;
                reci++;
            }
        }
        else
        {
            if(isalpha(c))
            {
                if(pocetak)
                    c = toupper(c);
                slova++;
            }

            if(poz==VAN_RECI)
                c = toupper(c);
            else
                c = tolower(c);
            poz = U_RECI;
        }
        putchar(c);
    }
    reci++; /*brojanje i poslednje reci*/

    /*stampa rezultata*/
    printf("\n\nBroj reci: %d", reci);
    printf("\n\nBroj slova: %d\n\n", slova);
}
```


Задатак 2_24.

Написати програм на језику C који:

- а) чита са тастатуре два низа *a* и *b* од по *n* ($0 \leq n \leq 20$) знакова,
- б) чита са тастатуре ред знакова и низ знакова *a* (или један његов део), који се појављују у тексту, замењује низом знакова *b* (или одговарајућим његовим делом) и приказује тако измењен ред знакова на екрану,
- в) понавља претходни корак све док не учита са тастатуре празан ред.

Решење 2_24.

```
#include <stdio.h>
#include <string.h>
#define MAX 20                                /*dozvoljen broj znakova u nizu*/

main()
{
    char a[MAX],b[MAX],red[80],j;
    int i,n;

    do{    printf("Unesite duzinu nizova a i b (n<20): ");
           scanf("%d",&n);
           fflush(stdin);    /*citanje zaostalih karaktera sa ulaza*/
        }while(n<1 || n>MAX);
    do
    {
        printf("\nUnesite niz znakova a: ");
        gets(a);
    }while(strlen(a)>MAX);
    do
    {
        printf("\nUnesite niz znakova b: ");
        gets(b);
    }while(strlen(b)>MAX);

    while(1)
    {
        printf("\nUnesite jedan red teksta:\n");
        gets(red);
        if(strlen(red)==0)
            break;

        printf("\nTekst sa odgovarajucim zamenama:\n");
        for(i=0;i<strlen(red);i++)
        {
            for(j=0;j<n;j++)
                if(red[i]==a[j])
                {
                    red[i]=b[j];
                    break;
                }
            putchar(red[i]);
        }
        putchar('\n');
    }
}
```

Задатак 2_25.

Написати програм на језику C који:

- а) чита са тастатуре ред знакова,
- б) формира низове знакова *niz1* и *niz2*, тако што сваки знак задатог реда текста, који је слово или цифра, уписује наизменично у *niz1* или *niz2*,
- в) формира низ знакова *niz* од елемената низа *niz1*, а после њих и од елемената низа *niz2*,
- г) понавља поступак (метод криптовања познат као транспозиција решетка) све док не учита празан ред.

Решење 2_25.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 80                /*broj znakova u jednom redu*/

main()
{
    char red[MAX], niz1[MAX/2], niz2[MAX/2], niz[MAX];
    int i,j,k;

    while(1)
    {
        printf("\nUnesite red teksta:\n");
        gets(red);

        if(strlen(red)==0)
            break;

        j=k=0;
        /*formiranje nizova niz1 i niz2*/
        for(i=0;i<strlen(red);i++)
        {
            if(isalnum(red[i]))
            {
                if(j<k)
                    niz2[j++]=red[i];
                else
                    niz1[k++]=red[i];
            }
        }
        niz1[k]='\0'; niz2[j]='\0';

        /*formiranje niza niz*/
        strcpy(niz,niz1);    /*kopira se niz1 u niz, od pocetka*/
        strcat(niz,niz2);    /*kopira se niz2 u niz, u produzetku sadrzaja*/

        printf("\nZadati red teksta:\n\t%s\n",red);
        printf("\nKodirani red teksta:\n\t%s\n\n", niz);
    }
}
```

Задатак 2_26.

Написати програм на језику C који:

- а) чита са тастатуре n ($n \leq 20$) редова текста (низова знакова),
- б) формира и приказује на екрану низ од непразних задатих редова текста, од којих сваки има мање од 30 знакова,
- в) приказује на екрану низ формиран под б) на следећи начин: свака децимална цифра замењена је једним знаком доња црта (_),
- г) приказује на екрану редове текста формиране под в), поређане по дужини од најкраћег.

Решење 2_26.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define BROJ      20          /*dozvoljen broj redova*/
#define MAX       80          /*broj znakova u jednom redu*/
#define MAX2      30          /*trazena duzina reda*/

main()
{
    int i, j=0, n, l;
    char pom[MAX+1], stringovi[BROJ][MAX+1],
          stringovi2[BROJ][MAX+1], stringovi3[BROJ][MAX+1];

    do
    {
        printf("\nBroj redova? ");
        scanf("%d", &n);
        fflush(stdin);          /*citanje zaostalih karaktera sa ulaza*/
    }while(n<1 || n>BROJ);

    /*formiranje drugog niza od ucitanog niza redova trazene duzine*/
    for(i=0; i<n; i++)
    {
        gets(stringovi[i]);      /*citanje jednog reda (niza znakova)*/
        l = strlen(stringovi[i]); /*izracunavanje duzine reda*/

        if(l>0 && l<MAX2)
            strcpy(stringovi2[j++], stringovi[i]);    /*kopiranje niza*/
    }
    n=j;

    /*formiranje treceg niza redova zamenom decimalnih cifara*/
    for(i=0; i<n; i++)
        for(j=0; j<MAX; j++)
            if(isdigit(stringovi2[i][j])) /*ako je znak decimalna cifra*/
                stringovi3[i][j]='_';    /*izmena znaka*/
            else
                stringovi3[i][j]=stringovi2[i][j]; /*kopiranje znaka*/
}
```

```

/*dovodjenje niza redova u neopadajuci poredak njihovih duzina*/
for(i=0;i<n-1;i++)
    for(j=i+1;j<n;j++)
        if(strlen(stringovi3[i])>strlen(stringovi3[j]))
        {
            strcpy(pom,stringovi3[i]);
            strcpy(stringovi3[i],stringovi3[j]);
            strcpy(stringovi3[j],pom);
        }

/*stampanje rezultata*/
printf("\nNovi niz:\n");
for(i=0; i<n; i++)
    puts(stringovi3[i]);          /*prikaz jednog reda (niza znakova)*/
}

```

Задатак 2_27.

Написати програм на језику C који:

- чита са тастатуре низ од n ($n \leq 10$) редова текста (низова знакова),
- приказује на екрану сваки задати ред који почиње целим бројем,
- инвертује места редовима у задатом низу (заменаује места првом и последњем, другом и претпоследњем...) и на екрану приказује резултат.

Решење 2_27.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define BROJ 10          /*dozvoljen broj redova*/
#define MAX 80           /*broj znakova u jednom redu*/

main()
{
    char redovi[BROJ][MAX+1], pom[MAX+1];
    int i, j=0, n;

    printf("\nBroj redova: ");
    scanf("%d", &n);getchar();

    /*citanje najvise n redova*/
    printf("\nZadati tekst od najvise %d redova:\n", n);
    for(i=0; i<n; i++)
        gets(redovi[i]);

    printf("\nRedovi koji pocinju celim brojem:\n");
    for(i=0; i<n; i++)
        if(atoi(redovi[i])!=0)          /*ako red pocinje celim brojem*/
            puts(redovi[i]);
}

```

```

printf("\nInvertovani niz redova:\n");
for(i=0; i<n/2; i++)          /*do polovine niza*/
{
    strcpy(pom, redovi[i]);    /*zamena mesta 1. i n-og, 2 i (n-1)-og...*/
    strcpy(redovi[i], redovi[n-1-i]);
    strcpy(redovi[n-1-i], pom);
}

for(i=0; i<n; i++)
    puts(redovi[i]);
}

```

Задатак 2_28.

Написати програм на језику C који:

- прихвата са тастатуре n ($n \leq 10$) редова текста, или док се не зада ред који се састоји само од знака тачка (.),
- приказује на екрану само оне редове из задатог текста који имају децималне цифре,
- приказује на екрану ред из задатог текста који има највећи број децималних цифара.

Решење 2_28.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define BROJ 10          /*dozvoljen broj redova*/
#define MAX 80          /*broj znakova u nazivima*/

main()
{
    int i, j, n, max=0, imax=-1, brcif;
    char red[BROJ][MAX+1];

    do
    {
        printf("Unesite broj redova teksta: ");
        scanf("%d",&n);
    }while(n<1||n>BROJ);

    printf("\nUnesite tekst:\n");
    j=getchar();

    /*citanje n redova teksta*/
    for(i=0;i<n;i++)
    {
        gets(red[i]);
        /*ako red ima samo znak tacka, kraj petlje*/
        if(strcmp(red[i],".")==0 )
            break;
    }
}

```

```

n=i;                                /*n je duzina niza redova*/
printf("\n");

for(i=0;i<n;i++)                    /*za svaki red u nizu*/
{
    brcif=0;

    for(j=0;j<strlen(red[i]);j++)    /*za svaki znak u redu*/
        if(isdigit(red[i][j]))      /*provera da li je decimalna cifra*/
            brcif++;

    if(brcif!=0)                    /*prikaz reda koji ima decimalne cifre*/
        puts(red[i]);

    if(brcif>max)                  /*odredjivanje reda sa najvise decimalnih cifara*/
    {
        max=brcif;
        imax=i;
    }
}
if(imax!=-1)
    printf("\nRed sa najvise cifara je:\n%s\n",red[imax]);
else
    printf("\nU tekstu nije bilo redova sa ciframa.\n");
}

```

Задатак 2_29.

Написати програм на језику C који:

- чита са тастатуре један ред знакова,
- приказује на екрану сваки задати ред знакова кодиран на следећи начин: после сваке цифре која се појављује најчешће од свих цифара у том реду додаје знак двотачка (:), као и сам број појављивања те цифре (ако има више оваквих цифара узима се у обзир прва од њих у задатом реду),
- понавља кораке под а) и б) све док се не унесе 20 редова, или ред у коме је најчешће појављивана цифра 0, или ред у коме уопште нема цифара.

Решење 2_29.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define BROJ 20                      /*dozvoljen broj redova*/
#define MAX 80                      /*broj znakova u jednom redu*/

main()
{
    char red[MAX+1], red2[MAX+1];
    int cifre[10];
    int i, j, k, c, max, ciframax, broj_ciframax;

```

```

/*citanje i kodiranje redova*/
for(i=0;i<BROJ;i++)
{
    /*inicijalizacija brojeva pojavljivanja pojedinih cifara*/
    for(j=0; j<10; j++)
        cifre[j]=0;

    k=0;

    /*citanje jednog reda znakova*/
    puts("\nUnesite red teksta");
    gets(red);

    for(j=0; j<strlen(red); j++)
    {
        c = red[j];
        if(isdigit(c))
            ++cifre[c-'0'];
    }

    max = cifre[0];
    ciframax=0;
    for(j=0; j<10; j++)
    {
        if(cifre[j]>max)
        {
            max = cifre[j];
            ciframax = j;
        }
    }

    if(ciframax==0)
        break;

    broj_ciframax=0;
    for(j=0; j<strlen(red); j++)
    {
        if((red[j]-'0')==ciframax)
            broj_ciframax++;
    }

    /*formiranje i prikaz kodiranog reda*/
    for(j=0; j<strlen(red); j++)
    {
        red2[k++]=red[j];
        if(red[j]==ciframax+'0')
        {
            red2[k++]='.';
            red2[k++]=broj_ciframax+'0';
        }
    }
    /*kada se string formira znak po znak, neophodna je i oznaka za kraj*/
    red2[k]='\0';
    puts(red2);
}

```

Задатак 2_30.

Написати програм на језику C који:

- а) чита са тастатуре један ред знакова,
- б) приказује на екрану задати ред знакова на следећи начин: сваку реч у посебном реду,
- в) одређује и приказује на екрану слово које се најчешће појављује у том реду,
- г) понавља кораке под а), б) и в) све док учитани ред садржи бар две речи.

Решење 2_30.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define BROJ_S 26 /*broj slova abecede (bez razlike velika/mala slova)*/
#define BROJ 30 /*dozvoljen broj redova*/
#define MAX 80 /*broj znakova u jednom redu*/

main()
{
    int i, c, broj_reci, max, max_slovo;
    int slova[BROJ_S];
    char red[MAX+1];

    while(1)
    {
        /*inicijalizacija brojeva pojavljivanja pojedinih slova*/
        for(i=0; i<BROJ_S;i++)
            slova[i]=0;
        broj_reci=1;

        /*citanje reda znakova*/
        puts("\nUnesite red teksta");
        gets(red);

        /*izdvajanje reci i kodiranje reda*/
        printf("\nZadate reci, svaka u posebnom redu:\n");
        for(i=0; i<strlen(red); i++)
        {
            c = red[i];

            if(c==' ' || c=='\t')
            {
                putchar('\n');
                broj_reci++;
            }
            else
            {
                putchar(c);
                if(isalpha(c))
                {
                    c=toupper(c);
                    slova[c-'A']++;
                }
            }
        }
    }
}
```



```

        if(broj_reci<2)                /*ako red nema bar dve reci, kraj*/
            break;

        max=slova[0];                  /*pretpostavka: najvise slova A*/
        max_slovo='A';
        for(i=0; i<BROJ_S;i++)
            if(slova[i]>max)            /*nadjeno je slovo koje se cesce pojavljuje*/
            {
                max=slova[i];
                max_slovo = i+'A';
            }
        printf("\n\nBroj reci u zadatom redu: %d", broj_reci);
        printf("\nNajvise se pojavljuje slovo: %c\n", max_slovo);
    }
}

```

Задатак 2_31.

Написати програм на језику C који:

- а) приказује на екрану децималне цифре, мала или велика слова абетеде из опсега чије границе (са знаком – у средини) добија из знаковног низа са тастатуре. (На пример ради:
- | | |
|--------------|--------------|
| са тастатуре | на екран |
| 3-5 | 345 |
| a-f | abcdef |
| C-M | CDEFGHIJKLM) |
- б) понавља корак под а) све док учитани знаковни низ има тачно три знака и на месту другог знака црту на средини (-). Ако на месту првог и трећег знака није децимална цифра, велико или мало слово, у тој итерацији петље знаковни низ не узима се у обзир.

Решење 2_31.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MAX 3                /*trazeni broj znakova u zadatom stringu*/

main()
{
    char s[MAX+1];
    int i, l;

    while(1)
    {
        printf("\n\nZadati u stringu granice opsega ");
        printf("\n(slovo-slovo ili cifra-cifra): ");
        gets(s);

        /*ako nije zadat string trazene duzine, kraj*/
        if(strlen(s)!=MAX || s[1]!='-')
            break;
    }
}

```

```

/*opseg vrednosti od 1. do poslednjeg znaka*/
l = s[2]-s[0];

/*ako nije zadat string trazenog oblika, ignorise se*/
if(! ( isupper(s[0]) && isupper(s[2]) && l>0 ||
        islower(s[0]) && islower(s[2]) && l>0 ||
        isdigit(s[0]) && isdigit(s[2]) && l>0      ) )
    continue;

/*prikaz svih znakova iz opsega zadatog pomocu stringa*/
for(i=0; i<=l; i++)
    putchar(s[0]+i);
}
}

```

Задатак 2_32.

Написати програм на језику C који:

- а) приказује на екрану слова абецедe од задатог до последњег, као и децималне цифре од задате до последње. Почетни слово и цифра добијају се из знаковног низа са тастатуре;

(На пример ради:	са тастатуре	на екран
	<i>s7</i>	<i>stuvwxyz789</i>
	<i>M3</i>	<i>MNOPQRSTUVWXYZ3456789)</i>

- б) понавља корак под а) све док учитани знаковни низ има тачно два знака. Ако на месту првог знака није слово, а на месту другог децимална цифра, у тој итерацији петље знаковни низ се не узима у обзир.

Решење 2_32.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MAX 2 /*dozvoljen broj znakova u trazenom stringu*/

main()
{
    char s[MAX+1];
    int i, l1, l2;

    while(1)
    {
        printf("\n\nZadati u stringu pocetno slovo i pocetnu cifru: ");
        gets(s);

        /*ako nije zadat string trazene duzine, kraj*/
        if(strlen(s)!=MAX)
            break;
    }
}

```

```

/*ako nije zadat string trazenog oblika, ignorise se*/
if( !isalpha(s[0]) || !isdigit(s[1]) )
    continue;

/*opseg vrednosti od 1. do poslednjeg znaka (za slova i brojeve)*/
l1 = (isupper(s[0]))? ('Z'-s[0]) : ('z'-s[0]);
l2 = '9'- s[1];

/*prikaz svih znakova iz opsega zadatog pomocu stringa*/
for(i=0; i<=l1; i++)
    putchar(s[0]+i);
for(i=0; i<=l2; i++)
    putchar(s[1]+i);
}
}

```

Задатак 2_33.

Написати програм на језику С који:

- чита са тастатуре датум (редни број месеца у години, као и годину, у облику *mm:gggg*),
- формира низ карактера *datumi* у који уписује назив месеца (задатог редног броја) и задату годину (на пример за учитане податке 06:2006 уписује у стринг *Jun2006*),
- чита са тастатуре n ($n \leq 10$) нових датума и сваки датум који је различит од првог задатог уписује у стринг *datumi*, у продужетку постојећег садржаја. Приказује на екрану садржај резултујућег стринга, као и најстарију годину из низа задатих.

Решење 2_33.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define BROJ 10
#define MAX1 4
#define MAX2 8

main()
{
    char niz[][MAX1] = { "Jan", "Feb", "Mar", "Apr", "Maj", "Jun",
                        "Jul", "Avg", "Sep", "Okt", "Nov", "Dec"};

    char god1[MAX1+1], god2[MAX1+1], datumi[(BROJ+1)*MAX2];
    int i, mes1, mes2, godine[BROJ+1], min_god;

    /*Citanje prvog zadatog datuma*/
    puts("\nUnesite datum: ");
    scanf("%d:%s", &mes1, god1);
}

```

```

godine[0] = atoi(god1);           /*dobijanje celog broja iz stringa*/

/*formiranje prvog dela stringa
 i prvog elementa niza godina*/
strcpy(datumi, niz[mes1-1]);
strcat(datumi, god1);

/*Citanje ostalih datuma, dodavanje istih stringu
 i formiranje ostalih elemenata niza godina*/
for(i=0; i<BROJ; i++)
{
    puts("\nUnesite nov datum: ");
    scanf("%d:%s", &mes2, god2);

    if(mes1!=mes2 || strcmp(god1,god2)!=0)
    {
        strcat(datumi, " ");
        strcat(datumi, niz[mes2-1]);
        strcat(datumi, god2);
        godine[i+1] = atoi(god2);
    }
}

/*nalazenje najstarije godine*/
min_god = godine[0];
for(i=0; i<BROJ+1; i++)
    if(godine[i]<min_god)
        min_god = godine[i];

/*stampanje rezultata*/
puts("\nFormirani datumi:");
puts(datumi);

printf("\n\nNajstarija godina u nizu: %d", min_god);
}

```

Задатак 2_34.

Написати програм на језику C који:

- чита са тастатуре податке за n ($n \leq 10$) испита (назив предмета, дан у месецу и месец у следећем облику: *naziv dd.mm*),
- формира низ стрингова *ispiti* у који уписује све задате називе предмета редом, као и низове *dani* и *meseci* у које уписује одговарајуће податке за сваки испит редом,
- приказује на екрану, опционо (за изабрану вредност 1/0), податке о испитима који су заказани од задатог датума (*dan* и *mesec*), или списак свих датума за предмет са траженим називом.

Решење 2_34.

```
#include <stdio.h>
#include <string.h>
#define BROJ 10          /*dozvoljen broj ispita*/
#define MAX 30           /*broj znakova u nazivu ispita*/

main()
{
    char predmet[MAX+1],ispiti[BROJ][MAX+1];
    int dan, mes, dani[BROJ], meseci[BROJ];
    int i, opcija;

    puts("\nZadati: naziv, dan i mesec za svaki predmet:");
    puts("\n(u obliku: naziv dd.mm)\n");

    for(i=0; i<BROJ; i++)
        scanf("%s%d.%d", ispiti[i], &dani[i], &meseci[i]);

    /*Izbor pretrage i stampanje rezultata*/
    printf("\nPretraga: po danu/po nazivu predmeta (1/0) ");
    scanf("%d", &opcija);

    if(opcija)                /*izabrana pretraga po danu*/
    {
        printf("\nRedni broj dana u mesecu i mesec (dd.mm): ");
        scanf("%d.%d", &dan, &mes);

        printf("\nSvi ispiti od zadatog datuma:\n");
        for(i=0; i<BROJ; i++)
        {
            if(dani[i]>=dan && meseci[i]>=mes)
                puts(ispiti[i]);
        }
    }

    else                       /*izabrana pretraga po nazivu predmeta*/
    {
        printf("\nIme predmeta: ");
        scanf("%s", predmet);

        printf("\nSvi datumi za trazeni ispit:\n");
        for(i=0; i<BROJ; i++)
        {
            if(strcmp(ispiti[i], predmet)==0)
                printf("\n%02d.%02d", dani[i], meseci[i]);
        }
    }
}
```

Задатак 2_35.

Написати програм на језику C који:

- а) чита са тастатуре низ знакова (број телефона) задат у посебном реду, у облику: 011-471-365 (први знак црта на средини (-) уноси се након 3 цифре, док су позиција другог знака црта на средини и број осталих цифара непознати),
- б) одваја:
 - прве три цифре у цео број: *pozivni_broj*,
 - остале цифре (без знака црта на средини) у стринг: *broj_telefona* и конвертује формирану стринг у одговарајући број,
- в) понавља корак под а) и б) *n* пута ($n \leq 20$), а онда приказује на екрану само оне задате бројеве телефона који имају тражени *pozivni_broj* (задат преко тастатуре), поређане у неопседајући поредак.

Решење 2_35.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define BROJ      20          /*dozvoljen broj unosa brojeva*/
#define MAX      80          /*broj znakova u jednom redu*/

main()
{
    int i,j, k, n, pom;
    int pozivni[BROJ], trazeni, niz[BROJ];
    char string[BROJ][MAX+1], string2[BROJ][MAX+1];

    do
    {
        printf("\nKoliko ce biti brojeva? ");
        scanf("%d", &n);
    }while(n<1 || n>BROJ);

    /*citanje brojeva u predvidjenom obliku i formiranje stringova*/
    printf("\nBrojevi u obliku 0xx-xxx-xxx:\n");
    for(i=0; i<n; i++)
    {
        scanf("%d-%s", &pozivni[i], string[i]);

        for(j=0,k=0; j<strlen(string[i]); j++)
            if(string[i][j] != '-')                /*ignorisanje znaka crta*/
                string2[i][k++] = string[i][j];
        string2[i][k] = '\0';                      /*oznaka za kraj stringa*/
    }

    /*trazeni pozivni broj*/
    puts("Trazeni pozivni broj?");
    scanf("%d", &trazeni);

    /*formiranje niza sa brojevima telefona*/
    for(i=0,j=0; i<n; i++)
```

```

    {      if(pozivni[i]==trazeni)
            niz[j++] = atoi(string2[i]);
    }
    n = j;                                     /*duzina formiranog niza*/

    /*formiranje i prikaz na ekranu sortiranog niza brojeva telefona*/
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(niz[i]>niz[j])
            {
                pom = niz[i];
                niz[i] = niz[j];
                niz[j] = pom;
            }

    for(i=0; i<n; i++)
        printf("\n%d", niz[i]);
}

```

Задатак 2_36.

Написати програм на језику C који:

- чита са тастатуре низ знакова (датум) задат у посебном реду, у облику: 2006/08/30,
- одваја:
 - прве четири цифре у цео број: *godina*,
 - остале цифре (без знака коса црта) у стринг: *datum*
 и конвертује формирани стринг у одговарајући број,
- понавља корак под а) и б) n пута ($n \leq 30$), а онда приказује на екрану само оне задате датуме (месец и дан у облику: 08/30) који припадају траженом броју *godina* (задатом преко тастатуре), поређане од последњег датума (у нарастући поредак).

Решење 2_36.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define BROJ      30          /*dozvoljen broj datuma*/
#define MAX      80          /*broj znakova u redu*/

main()
{
    int i,j, k, n, pom;
    int godina[BROJ], trazena, niz[BROJ];
    char string[BROJ][MAX+1], string2[BROJ][MAX+1];

    do
    {
        printf("\nKoliko ce biti datuma? ");

```

```

        scanf("%d", &n);
    }while(n<1 || n>BROJ);

    /*citanje datuma u predvidjenom obliku i formiranje novog stringa*/
    printf("\nDatumi u obliku gggg/mm/dd:\n");
    for(i=0; i<n; i++)
    {
        scanf("%d/%s", &godina[i], string[i]);
        for(j=0,k=0; j<strlen(string[i]); j++)
            if(isdigit(string[i][j])) /*ignorisanje znakova koji nisu cifre*/
                string2[i][k++] = string[i][j];
        string2[i][k] = '\0'; /*oznaka za kraj stringa*/
    }

    puts("Godina?"); scanf("%d", &trazena);

    /*formiranje niza brojeva od stringova*/
    for(i=0,j=0; i<n; i++)
    {
        if(godina[i]==trazena)
            niz[j++] = atoi(string2[i]);
    }
    n = j; /*duzina formiranog niza*/

    /*formiranje i prikaz na ekranu sortiranog niza datuma*/
    printf("\nTrazeni niz datuma u nerastucem poretku:\n");
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(niz[i]<niz[j])
            {
                pom = niz[i];
                niz[i] = niz[j];
                niz[j] = pom;
            }
    for(i=0; i<n; i++)
        printf("\n%02d/%02d", niz[i]/100, niz[i]%100);
}

```


3. Функције, показивачи, динамичка додела меморије и структуре у програмима на језику C

Задатак 3_1.

Написати програм на језику C који:

- а) чита са тастатуре низ у коме је спаковано више низова целих бројева на следећи начин: прво дужина низа а онда и сви елементи низа, крај уноса означава дужина низа $n \leq 0$,
- б) динамички додељује меморију, формира и приказује на екрану нови низ, који за сваки низ учитан под а) добија по један број: 1 (ако је највише позитивних или нула елемената), -1 (ако је највише негативних) или 0 (ако је исти број позитивних (и нула) и негативних елемената).

Решење 3_1.

```
#include <stdio.h>
#include <stdlib.h>

void cita(int niz[], int n);
void stampa(int niz[], int n);
int poz_neg(int niz[], int n);

main()
{
    int n, n2=0;
    int *niz, *novi;

    /*Citanje i izdvajanje nizova iz jednog zadatog*/
    printf("\nZadati niz spakovanih nizova:\n");
    while(1)
    {
        scanf("%d", &n);
        if(n<=0) break;

        /*dodela memorije za svaki izdvojeni niz*/
        niz = malloc(n*sizeof(int));
        if(niz==NULL) exit(1);

        /*izdvajanje jednog niza u pomocni niz*/
        cita(niz, n);

        if(!n2) /*za prvi izdvojeni niz*/
        {
            novi=malloc(sizeof(int));
            if(novi==NULL) exit(1);
        }
        else /*za svaki sledeci izdvojeni niz*/
        {
            novi=realloc(novi, (n2+1)*sizeof(int));
            if(novi==NULL) exit(1);
        }
        /*kreiranje novog elementa niza y*/
        novi[n2++] = poz_neg(niz, n);
        /*oslobadjanje memorije za pomocni niz*/
        free(niz);
    }
}
```

```

        /*stampanje rezultata*/
        stampa(novi, n2);

        /*oslobadjanje memorije za novi niz*/
        free(novi);

        return 0;
    }
void cita(int niz[], int n)
{
    int i;

    for(i=0; i<n; i++)
        scanf("%d ", &niz[i]);
}
int poz_neg(int niz[], int n)
{
    int i, broj_poz=0, broj_neg=0;

    for(i=0; i<n; i++)
    {
        if(niz[i]>=0)
            broj_poz++;
        else
            broj_neg++;
    }
    return( (broj_poz==broj_neg)? 0 : ( (broj_poz>broj_neg)? 1: -1) );
}
void stampa(int niz[], int n)
{
    int i;

    printf("\nNovi niz:\n");
    for(i=0; i<n; i++)
        printf("%d ", niz[i]);
}

```

Задатак 3_2.

Написати програм на језику C који:

- чита са тастатуре дужину низа n , динамички додељује меморију за низ од n целих бројева и учитава ове бројеве са тастатуре, крај уноса низова означава дужина низа $n \leq 0$, или највише 10 задатих низова,
- мења сваки задати низ тако што: ако је дужи од првог задатог одбацује вишак елемената, ако је краћи од првог задатог додаје нуле до дужине првог низа и приказује на екрану тако измењен низ,
- формира и приказује на екрану низ од бројева одбачених елемената из свих задатих низова редом (ако је из првог низа одбачено 5 елемената први елемент новог низа има вредност 5...).

Решење 3_2.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10 /*dozvoljen broj nizova*/

void cita(int niz[], int n);
void menja(int niz[], int n1, int n2, int novi[], int novih);
void stampa(int niz[], int n);

main()
{
    int *niz1, *niz2, *novi;
    int i, n1, n2, rbr=2, broj=0;
    int novih=0; /*duzina niza odbacenih elemenata*/

    /*dodela memorije za niz brojeva odbacenih elemenata*/
    novi = malloc(MAX*sizeof(int));
    if(novi==NULL)
        exit(1);

    /*dodela memorije i citanje prvog od nizova*/
    do
    {
        printf("\nDuzina 1. niza? ");
        scanf("%d", &n1);
    }while(n1<=0);

    niz1 = malloc(n1*sizeof(int));
    if(niz1==NULL)
        exit(1);

    printf("Elementi 1. niza: ");
    cita(niz1, n1);

    /*dodela memorije, citanje, izmena i stampa ostalih nizova*/
    for(i=0; i<MAX; i++)
    {
        printf("\n\nDuzina %d. niza? ", rbr);
        scanf("%d", &n2);

        if(n2<=0)
            break;

        /*svaki niz treba da bude duzine iste kao i prvi*/
        niz2 = malloc(n2*sizeof(int));
        if(niz2==NULL)
            exit(1);

        printf("Elementi %d. niza: ", rbr++);
        cita(niz2, n2);

        niz2 = realloc(niz2, n1*sizeof(int));
```

```

        if(niz2==NULL)
            exit(1);

        menja(niz2, n1, n2, novi, novih);
        printf("Izmenjen niz: ");
        stampa(niz2, n1);

        if(n1<n2)                /*ako je bilo odbacivanja elemenata*/
            novih++;             /*niz novi[] dobija jos jedan element*/

        /*oslobadjanje memorije*/
        free(niz2);
    }

    printf("\nNiz od brojeva odbacenih elemenata: ");
    if(novih!=0)
        stampa(novi, novih);

    /*oslobadjanje memorije*/
    free(niz1);
    free(novi);

    return 0;
}

void cita(int niz[], int n)
{
    int i;
    for(i=0; i<n; i++)
        scanf("%d", &niz[i]);
}

void menja(int niz[], int n1, int n2, int novi[], int novih)
{
    int i;

    if(n2<n1)
        for(i=n2; i<n1; i++)
            niz[i] = 0;
    else if(n2>n1)
        novi[novih] = n2-n1;
}

void stampa(int niz[], int n)
{
    int i;

    for(i=0; i<n; i++)
        printf("%d ", niz[i]);
}

```

Задатак 3_3.

Написати програм на језику C који

- чита са тастатуре дужину низа n , динамички додељује меморију а затим чита са тастатуре два низа од по n реалних бројева,
- у сваком од учитаних низова одређује најмању вредност и одбацује све елементе чије су вредности једнаке најмањој у том низу,
- додаје све елементе другог низа на крај првог низа (низови формирани под б)) и приказује на екрану резултујући низ.

Решење 3_3.

```
#include <stdio.h>
#include <stdlib.h>

void cita(float niz[], int n);
float minimum(float niz[], int n);
int odbacuje(float niz[], int n, float min);
int dodaje(float niz1[], int n1, float niz2[], int n2);
void stampa(float niz[], int n);

main()
{
    float *niz1, *niz2;
    int n, n1, n2;

    /*dodela memorije i citanje nizova*/
    do
    {
        printf("\nDuzina svakog niza? "); scanf("%d", &n);
    }while(n<=0);

    niz1 = malloc(n*sizeof(float));
    if(niz1==NULL) exit(1);
    niz2 = malloc(n*sizeof(float));
    if(niz2==NULL) exit(1);

    printf("\nElementi 1. niza: "); cita(niz1, n);
    printf("\nElementi 2. niza: "); cita(niz2, n);

    /*dobijanje novih duzina nizova*/
    n1 = odbacuje(niz1, n, minimum(niz1, n));
    n2 = odbacuje(niz2, n, minimum(niz2, n));

    /*promena duzine prvog niza, da moze prihvatiti drugi*/
    niz1 = realloc(niz1, (n1+n2)*sizeof(float));
    if(niz1==NULL) exit(1);

    /*dodavanje drugog niza prvom i stampanje rezultata*/
    n1 = dodaje(niz1, n1, niz2, n2);
    stampa(niz1, n1);
}
```

```

        /*oslobadjanje memorije*/
        free(niz1);
        free(niz2);
        return 0;
    }

    void cita(float niz[], int n)
    {
        int i;
        for(i=0; i<n; i++)
            scanf("%f", &niz[i]);
    }

    float minimum(float niz[], int n)
    {
        int i;

        float min = niz[0];
        for(i=0; i<n; i++)
            if(niz[i]<min)
                min = niz[i];

        return min;
    }

    int odbacuje(float niz[], int n, float min)
    {
        int i;
        float *ptr1, *ptr2;
        ptr1 = ptr2 = niz;                                /*ptr1 i ptr2 cuvaju adresu istog niza*/

        for(i=0; i<n; i++)
        {
            if(*ptr1!=min)                                /*za element koji nije najmanje vrednosti*/
            {
                *ptr2 = *ptr1;                            /*ptr2 pokazuje na sadrzaj tog elementa*/
                ptr2++;                                    /*i inkrementira se na jednu adresu dalje*/
            }
            ptr1++;                                        /*ptr1 se inkrementira u svakom slucaju*/
        }
        return (ptr2-niz);                                /*funkcija vraca broj trazениh elemenata*/
    }

    int dodaje(float niz1[], int n1, float niz2[], int n2)
    {
        int i;

        for(i=0; i<n2; i++)
            niz1[n1+i] = niz2[i];
        return (n1+n2);
    }

    void stampa(float niz[], int n)
    {
        int i;

        printf("\nRezultujuci niz:\n");
        for(i=0; i<n; i++)
            printf("%f ", niz[i]);
    }

```

Задатак 3_4.

Написати програм на језику C који

- а) чита са тастатуре дужину низа n , динамички додељује меморију за низ и чита са тастатуре сам низ од n целих бројева,
- б) сажима задати низ на следећи начин: избацује из низа сваки елемент чија је вредност већа од средње вредности истог низа,
- в) одређује број међусобно различитих елемената низа добијеног под б),
- г) понавља све кораке док се задаје дужина низа $n > 0$.

Решење 3_4.

```
#include<stdio.h>
#include<stdlib.h>

void cita(int [],int);
int sazima(int [],int);
int razlicitih(int [],int);

main()
{
    int i, n, m, *niz;

    while(1)
    {
        printf("\nKoliko niz ima elemenata (0<n<=100)? ");
        scanf("%d",&n);
        if(n<=0)
            break;

        /*dodela memorije za niz*/
        niz=malloc(n*sizeof(int));
        if(niz==NULL) exit(1);

        cita(niz,n);
        n=sazima(niz,n);

        if(n!=0)
        {
            /*izmena velicine dodeljene memorije*/
            niz=realloc(niz,n*sizeof(int));
            if(niz==NULL) exit(1);

            printf("\nSazeti niz je:\n");
            {
                for(i=0;i<n;i++)
                    printf("%d ", niz[i]);
                printf("\n");
            }

            m=razlicitih(niz,n);
            printf("\nRazlicitih elemenata posle sazimanja: %d\n",m);
        }
        else printf("\nU sazetom nizu nema elemenata\n");
    }
}
```



```

        /*oslobadjanje memorije*/
        free(niz);
    }
    return 0;
}

void cita(int x[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d. clan: ",i+1);
        scanf("%d",&x[i]);
    }
}

int sazima(int x[],int n)
{
    int i, s=0;
    int *ptr1, *ptr2;
    ptr1 = ptr2 = x;                /*ptr1 i ptr2 cuvaju adresu istog niza*/

    for(i=0;i<n;i++)
        s+=x[i];
    s/=n;

    for(i=0; i<n; i++)
    {
        if(*ptr1>s)                /*za element veci od srednje vrednosti*/
        {
            *ptr2 = *ptr1;        /*ptr2 pokazuje na sadrzaj tog elementa*/
            ptr2++;                /*i inkrementira se na jednu adresu dalje*/
        }
        ptr1++;                    /*ptr1 se inkrementira u svakom slucaju*/
    }
    return (ptr2-x);                /*funkcija vraca broj trazениh elemenata*/
}

int razlicitih(int x[],int n)
{
    int i,j,nadjeni_isti, broj_razl=1;

    for(i=0; i<n-1; i++)
    {
        nadjeni_isti = 0;
        for(j=i+1; j<n; j++)
        {
            if(x[i]==x[j])
            {
                nadjeni_isti=1;
                break;
            }
        }
        if(!nadjeni_isti)
            broj_razl++;
    }
    return broj_razl;
}

```

Задатак 3_5.

Написати програм на језику C који

- а) чита са тастатуре дужину низа n , динамички додељује меморију за низ и чита са тастатуре сам низ од n целих бројева,
- б) сажима задати низ на следећи начин: оставља само елементе између прве и друге нуле задатог низа,
- в) ако задати низ нема две нуле приказује одговарајућу поруку на екрану.

Решење 3_5.

```
#include<stdio.h>
#include<stdlib.h>

void cita(int [],int);
void stampa(int [],int);
int nule(int [],int);

main()
{
    int n,p=0;
    int *niz;

    do
    {
        printf("\nKoliko niz ima elemenata ? ");
        scanf("%d",&n);
    }while(n<=0);

    /*dodela memorije za niz*/
    niz=malloc(n*sizeof(int));
    if(niz==NULL) exit(1);

    cita(niz,n);
    n=nule(niz,n);

    if(n>0)
        stampa(niz,n);
    else
        printf("\nZadati niz nema dve nule.\n");

    /*oslobadjanje memorije*/
    free(niz);
    return 0;
}

void cita(int x[],int n)
{
    int i;

    printf("\nUnesite %d elemenata niza\n", n);
    for(i=0;i<n;i++)
    {
        printf("\n%d. element: ",i+1);
```

```

        scanf("%d",&x[i]);
    }
}
void stampa(int x[],int n)
{
    int i;

    printf("\nElementi niza su: \n");
    for(i=0;i<n;i++)
        printf("%d ",x[i]);
    printf("\n\n");
}
int nule(int x[],int n)
{
    int i, prva=0;
    int *ptr1, *ptr2;

    ptr1 = ptr2 = x;                /*ptr1 i ptr2 cuvaju adresu istog niza*/

    for(i=0; i<n; i++)
    {
        if(x[i]==0)
            prva++;

        else if(prva==1)             /*ako je element izmedju 1. i 2. nule*/
        {
            *ptr2 = *ptr1;          /*ptr2 pokazuje na sadrzaj tog elementa*/
            ptr2++;                 /*i inkrementira se na jednu adresu dalje*/
        }
        ptr1++;                    /*ptr1 se inkrementira u svakom slucaju*/
    }
    if(prva>1)
        return (ptr2-x);            /*funkcija vraca novu duzinu niza*/
    else
        return 0;
}

```

Задатак 3_6.

Написати програм на језику C који:

- дефинише структуру типа *ispit* чији су чланови: *predmet*, *datum* (у облику: *xx.xx.xxxx.*) и *sati*,
- декларише низ *smer* од највише 30 структура, иницијализује 3 структуре у низу на вредности:


```

                Ispit1, 01.09.2006. , 10
                Ispit2, 05.09.2006. , 10
                Ispit3, 10.09.2006. , 10
            
```

 и приказује на екрану овај списак испита са назначеним редним бројевима,
- чита са тастатуре редни број испита и ако је редни број у опсегу постојећих ажурира вредности чланова *datum* и *sati*, а у супротном уписује све податке за нови испит (*predmet*, *datum*, *sati*) у следећу структуру у низу,
- приказује на екрану резултат измене података у структурама.

Решење 3_6.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 30          /*duzina naziva predmeta*/
#define BROJ 3          /*pocetni broj ispita*/

const char *nazivi[3] = {"Ispit1", "Ispit2", "Ispit3"};
const char *datumi[3] = {"01.09.2006.", "05.09.2006.", "10.09.2006."};

struct ispit{           /*definicija jednog tipa strukture*/
    char predmet[MAX+1];
    char datum[MAX+1];
    int sati;
};

void init(struct ispit niz[]);
int cita(struct ispit niz[], int broj_ispita);
void stampa(struct ispit niz[], int broj_ispita);

main()
{
    int broj_ispita = BROJ;    /*pocetni broj struktura u nizu*/
    struct ispit *smer;        /*deklaracija pokazivaca na niz struktura*/

    /*dodela memorije za BROJ struktura tipa ispit*/
    smer = malloc(BROJ*sizeof(struct ispit));
    if(smer==NULL) exit(1);

    /*inicijalizacija i prikaz pocetnog niza struktura*/
    init(smer);
    stampa(smer, BROJ);

    /*izmena niza struktura i prikaz rezultata*/
    broj_ispita = cita(smer, broj_ispita);
    stampa(smer, broj_ispita);

    /*oslobadjanje memorije*/
    free(smer);
    return 0;
}

void init(struct ispit niz[])
{
    int i;

    for(i=0; i<BROJ; i++)
    {
        strcpy(niz[i].predmet, nazivi[i]);
        strcpy(niz[i].datum, datumi[i]);
        niz[i].sati = 10;
    }
}
```

```

int cita(struct ispit niz[], int broj_ispita)
{
    int rbr;

    while(1)                /*sve dok se unose podaci*/
    {
        printf("\n\nRedni broj ispita: ");
        scanf("%d", &rbr);
        if(rbr<1)
            break;
        /*moze se dodati samo prva sledeca struktura*/
        if(rbr>=broj_ispita+1)
        {
            rbr=broj_ispita+1;
            printf("\nIme predmeta, datum i sati?\n");
            scanf("%s%s%d", niz[rbr-1].predmet,
                niz[rbr-1].datum,
                &niz[rbr-1].sati);

            broj_ispita++;
        }
        /*ili izmeniti postojeca u nizu*/
        else
        {
            printf("\nDatum i sati: ");
            scanf("%s%d", niz[rbr-1].datum, &niz[rbr-1].sati);
        }
        if(broj_ispita>MAX)
            break;
    }
    return broj_ispita;
}

void stampa(struct ispit niz[], int broj_ispita)
{
    int i;

    printf("\nSpisak ispita u roku:\n");
    for(i=0; i<broj_ispita; i++)
        printf("\n%d. %s %s %d", i+1, niz[i].predmet,
            niz[i].datum,
            niz[i].sati);
}

```

Задатак 3_7.

Написати програм на језику C који:

- креира списак књига, тако што га попуњава подацима задатим преко тастатуре за n књига: *naziv*, *autor*, *god_izdanja*, *tiraz*,
- чита са тастатуре опцију унету преко тастатуре G / T (*godina* / *tiraz*) и у зависности од изабране опције:
- приказује на екрану називе свих књига које су издате у години задатој преко тастатуре, или
- приказује на екрану називе свих књига чији је тираж био већи од задатог преко тастатуре.

Решење 3_7.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAX1 30                /*najveci broj znakova u nazivima*/

struct knjiga{                /*definicija tipa strukture*/
    char naziv[MAX1+1];
    char autor[MAX1+1];
    int god_izdanja;
    int tiraz;
};

void cita(struct knjiga niz[], int n);
void pretrazuje(struct knjiga niz[], int n);

main()
{
    int n;
    struct knjiga *knjige;      /*deklaracija pokazivaca na niz struktura*/

    printf("\nBroj knjiga? ");
    scanf("%d", &n);

    /*dodela memorije za niz struktura tipa knjiga*/
    knjige = malloc(n*sizeof(struct knjiga));
    if(knjige==NULL) exit(1);

    /*pretrazivanje ucitanog spiska i prikaz rezultata*/
    cita(knjige, n);
    pretrazuje(knjige, n);

    /*oslobadjanje memorije*/
    free(knjige);

    return 0;
}

void cita(struct knjiga niz[], int n)
{
    int i;

    printf("\nZa %d knjiga (svaki podatak u posebnom redu):", n);
    printf("\nNaziv Ime_ autora Godina_izdanja Tiraz?\n");

    for(i=0; i<n; i++)
    {
        printf("\n%d. knjiga:\n", i+1);
        scanf("%s%s", niz[i].naziv, niz[i].autor);
        scanf("%d%d", &niz[i].god_izdanja, &niz[i].tiraz);
    }
}
```

```

void pretrazuje(struct knjiga niz[], int n)
{
    int i, c, godina, tiraz;

    printf("\nIzaberi opciju: G / T (po godini izdanja / po tirazu): ");
    getchar(); c = getchar(); getchar();

    if(toupper(c)=='G')
    {
        printf("\nGodina izdanja?");
        scanf("%d", &godina);

        printf("\nSpisak knjiga sa traženom godinom izdanja:\n");
        for(i=0; i<n; i++)
            if(niz[i].god_izdanja==godina)
                printf("\n%s", niz[i].naziv);
    }
    else if(toupper(c)=='T')
    {
        printf("\nTiraz?");
        scanf("%d", &tiraz);

        printf("\nSpisak knjiga sa tirazom većim od zadatog:\n");
        for(i=0; i<n; i++)
            if(niz[i].tiraz>tiraz)
                printf("\n%s", niz[i].naziv);
    }
    else
        printf("\nNepoznata opcija!");
}

```

Задатак 3_8.

Написати програм на језику C који:

- креира списак који има 4 колоне: у првој се аутоматски генерише редни број датума у списку (*rbr*), друга је за редни број дана у месецу (*dan*), трећа је за редни број месеца у години (*mes*), а четврта је за годину (*god*),
- чита са тастатуре датуме у облику: *dd.mm.gggg.*, њих 10, или све док датум који се задаје има смисла (и задата година је од 1900. до 2006.),
- уређује списак (низ структура) по датумима од најновијег датума и приказује тако уређен списак на екрану.

Решење 3_8.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX1 10                                /*Broj datuma*/

```

```

struct datum{                                /*definicija tipa strukture*/
    int rbr, dan, mesec, god;
};
int cita(struct datum niz[]);
int korektan_datum(int d, int m, int g);
int broj_dana(int m, int g);
int prestupna(int g);
void uredjen_spisak(struct datum niz[], int n);

main()
{
    int n;
    struct datum *niz;                      /*deklaracija pokazivaca na niz struktura*/

    /*dodela memorije za niz struktura tipa datum*/
    niz = malloc(MAX1*sizeof(struct datum));
    if(niz==NULL) exit(1);

    /*citanje samo korektnih datuma*/
    n = cita(niz);

    /*ako je broj datuma manji od dozvoljenog*/
    if(n<MAX1)
    {
        /*promena velicine memorije*/
        niz = realloc(niz, n*sizeof(struct datum));
        if(niz==NULL) exit(1);
    }

    /*uredjivanje i stampanje spiska*/
    uredjen_spisak(niz, n);

    /*oslobadjanje memorije*/
    free(niz);
    return 0;
}

int cita(struct datum niz[])
{
    int i, n=0, dan, mesec, god;

    for(i=0; i<MAX1; i++)
    {
        printf("\nDatum (u obliku dd.mm.gggg.)?");
        printf("\n(za kraj uneti datum koji nema smisla): ");
        scanf("%d.%d.%d",&dan,&mesec,&god);

        if(korektan_datum(dan, mesec, god))    /*ako datum ima smisla*/
        {
            niz[n].dan = dan;                  /*formiranje clana strukture*/
            niz[n].mesec = mesec;
            niz[n++].god = god;
        }
        else
            break;
    }
}

```



```

        return n;                                /*funkcija vraca broj kreiranih struktura*/
    }
    int korektan_datum(int d, int m, int g)
    {
        if(      d>=1 && d<=broj_dana(m,g) &&
                m>=1 && m<=12 &&
                g>1900 && g<=2006                )
            return 1;
        else
            return 0;
    }
    int broj_dana(int m, int g)
    {
        int broj=0;

        switch(m)
        {
            case 1:case 3:case 5:case 7:case 8:case 10:case 12:
                broj = 31;
                break;
            case 4:case 6:case 9:case 11:default:
                broj = 30;
                break;
            case 2:
                broj=28+prestupna(g);
                break;
        }

        return broj;
    }
    int prestupna(int g)                            /*vraca 1 za prestupnu godinu*/
    {
        return(g%100!=0 && g%4==0 || g%400==0);
    }
    void uredjen_spisak(struct datum niz[], int n)
    {
        int i, j;
        struct datum temp;

        for(i=0; i<n-1; i++)                        /*sortiranje datuma od najnovijeg*/
            for(j=i+1; j<n; j++)
            {
                if(niz[i].god<niz[j].god ||
                   (niz[i].god==niz[j].god && niz[i].mesec<niz[j].mesec) ||
                   (niz[i].god==niz[j].god && niz[i].mesec==niz[j].mesec
                    && niz[i].dan<niz[j].dan)
                )
                {
                    temp = niz[i];                  /*zamena mesta strukturama u nizu*/
                    niz[i] = niz[j];
                    niz[j] = temp;
                }
            }
        for(i=0; i<n; i++)                          /*uredjivanje rednih brojeva*/
            niz[i].rbr = i+1;

        for(i=0; i<n; i++)
            printf("\n%d.\t%02d.%02d.%d.",
                  niz[i].rbr, niz[i].dan, niz[i].mesec, niz[i].god);
    }

```

Задатак 3_9.

Написати програм на језику C који:

- дефинише структуру која садржи податке о испиту: *redni_broj*, *ime_predmeta*, *dan*, *mesec*, *broj_studenata*,
- чита са тастатуре податке о испитима, њих 50, или све док датум испита припада неком од регуларних испитних рокова у једној години (редни број месеца: 1., 4., 6. или 9., а редни број дана у месецу одговарајући),
- уређује списак испита (низ структура) по датумима у сваком року (од најстаријег датума) и приказује тако уређен списак на екрану.

Решење 3_9.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX1 50          /*dozvoljen broj datuma*/
#define MAX2 10          /*broj znakova u nazivima*/

struct datum{            /*definicija tipa strukture*/
    int rbr;
    char ime[MAX2];
    int dan, mesec, broj_stud;
};

int cita(struct datum niz[]);
int korektan_datum(char *ime, int d, int m);
int broj_dana(int m);
void uredjen_spisak(struct datum niz[], int n);

main()
{
    int n;
    struct datum *niz;      /*deklaracija pokazivaca na niz struktura*/

    /*dodela memorije*/
    niz = malloc(MAX1*sizeof(struct datum));
    if(niz==NULL)
        exit(1);

    n = cita(niz);

    /*promena velicine dodeljene memorije*/
    niz = realloc(niz, n*sizeof(struct datum));
    if(niz==NULL)
        exit(1);

    /*uredjivanje i prikaz spiska*/
    uredjen_spisak(niz, n);
}
```

```

/*oslobadjanje memorije*/
free(niz);
return 0;
}

int cita(struct datum niz[])
{
    int n=0, m, dan, mesec, broj;
    char ime[MAX2];

    printf("\nBroj ispita: ");
    scanf("%d", &m);

    printf("\nZadati podatke za svaki ispit");
    printf("(u obliku: naziv_ispita dd.mm. broj_studenata)");
    printf("\n(mesec moze biti samo: 01. 04. 06. ili 09.)");

    while(1)
    {
        printf("\nIspit %d.: ", n+1);
        scanf("%s%d.%d. %d", ime, &dan, &mesec, &broj);

        if(korektan_datum(ime, dan, mesec))
        {
            niz[n].rbr = n+1;
            strcpy(niz[n].ime, ime);
            niz[n].dan = dan;
            niz[n].mesec = mesec;
            niz[n++].broj_stud = broj;          /*inkrement broja struktura*/

            if(n==m)
                break;
        }
        else
            continue;
    }

    return n;                                /*funkcija vraca broj kreiranih struktura*/
}

int korektan_datum(char *ime, int d, int m)
{
    if((m==1 || m==4 || m==6 || m==9) && d>=1 && d<=broj_dana(m))
        return 1;
    else
        return 0;
}

int broj_dana(int m)
{
    int broj=0;
    switch(m)                                /*dozvoljeni su redni brojevi samo nekih meseci*/
    {
        case 1:
            broj = 31;
            break;
        case 4:case 6:case 9:
            broj = 30;
            break;
    }
}

```

```

        default:
            break;
    }
    return broj;
}

void uredjen_spisak(struct datum niz[], int n)
{
    int i, j;
    struct datum temp;

    printf("\nSortirani datumi od najstarijeg:\n");
    for(i=0; i<n-1; i++) /*sortiranje datuma*/
        for(j=i+1; j<n; j++)
        {
            if( niz[i].mesec>niz[j].mesec ||
                niz[i].mesec==niz[j].mesec && niz[i].dan>niz[j].dan )
            {
                temp = niz[i];
                niz[i] = niz[j];
                niz[j] = temp;
            }
        }

    for(i=0; i<n; i++) /*uredjivanje rednih brojeva*/
        niz[i].rbr = i+1;

    for(i=0; i<n; i++)
        printf("\n%d. %s %02d.%02d. %d", niz[i].rbr, niz[i].ime,
            niz[i].dan,niz[i].mesec, niz[i].broj_stud );
}

```

Задатак 3_10.

Написати програм на језику C који:

- дефинише структуру која садржи податке о стану: *ulica*, *broj*, *broj_soba*, *kvadrata*,
- чита са тастатуре податке о становима (њих највише 300) и креира списак све док за вредности чланова *broj_soba* и *kvadrata* важи: *broj_soba* је бар 5, а *kvadrata* у опсегу од 30 до 300,
- уређује и приказује на екрану податке о свим становима који се налазе у улици са задатим именом.

Решење 3_10.

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX1 5 /*trazeni broj soba*/
#define MAX2 30 /*minimalni broj kvadrata*/
#define MAX3 300 /*maksimalni broj kvadrata*/

```

```

struct stan{                                /*definicija tipa strukture*/
    char ulica[MAX2];
    int broj,broj_soba,kvadrata;
};

int cita(struct stan niz[]);
int korektan_broj(int soba,int kvadrata);
void spisak_ulica(struct stan niz[],int n);

main()
{
    struct stan *niz;                        /*deklaracija pokazivaca na strukturu*/
    int broj_stanova;

    /*dodela memorije*/
    niz=malloc(MAX3*sizeof(struct stan));
    if(niz==NULL) exit(1);

    broj_stanova = cita(niz);

    /*promena velicine dodeljene memorije*/
    niz = realloc(niz, broj_stanova*sizeof(struct stan));
    if(niz==NULL) exit(1);

    /*pretraga i prikaz formiranog spiska*/
    spisak_ulica(niz,broj_stanova);

    /*oslobadjanje memorije*/
    free(niz);
    return 0;
}

int cita(struct stan niz[])
{
    char ulica[MAX2];
    int n=0, broj, broj_soba, kvadrata;

    printf("\nUnesite podatke o stanu: ");
    printf("ulica broj broj_soba broj kvadrata.\n");
    printf("(broj_soba<=%d i %d<=kvadrata<=%d)\n", MAX1, MAX2, MAX3);
    printf("(za kraj uneti vrednosti van ovih opsega):\n\n");

    while(1)
    {
        scanf("%s%d%d%d", ulica, &broj, &broj_soba, &kvadrata);

        if(korektan_broj(broj_soba,kvadrata))
        {
            strcpy(niz[n].ulica, ulica);
            niz[n].broj=broj;
            niz[n].broj_soba = broj_soba;
            niz[n++].kvadrata = kvadrata;    /*inkrement broja struktura*/
        }
        else break;
    }
}

```

```

        return n;                                /*funkcija vraca broj kreiranih struktura*/
    }
int korektan_broj(int soba,int kvadrata)
{
    if(soba>0 && soba<=MAX1 && kvadrata>=MAX2 && kvadrata<=MAX3)
        return 1;
    else
        return 0;
}
void spisak_ulica(struct stan niz[],int n)
{
    int i;
    char ulica[MAX2];

    while(1)
    {
        printf("\nUnesite ime ulice za pretragu");
        printf("\n(znak za novi red je oznaka za kraj unosa): ");
        getchar();                                /*zaostali znak za prelaz u novi red*/
        gets(ulica);                              /*citanje imena za pretragu*/

        if(strlen(ulica)==0)
            break;
        else
            for(i=0;i<n;i++)
                if(!strcmp(niz[i].ulica, ulica))
                    printf("\n%s %d, %d soba %d kvadrata",
                        niz[i].ulica, niz[i].broj,
                        niz[i].broj_soba,niz[i].kvadrata);
    }
}

```

4. Аргументи из командне линије и рад са датотекама у програмима на језику С

Задатак 4_1.

Написати програм на језику C који:

- а) добија из командне линије имена од две текстуалне датотеке,
- б) копира садржај прве датотеке у другу и на екран тако да сви редови текста буду исте дужине као најдужи ред у тексту (редове који су краћи од најдужег допуњује на крају знаковима ^),
- в) ако из командне линије пре имена датотека добије опцију у облику *-num* решава задатак под б) тако што додаје на почетак сваког реда број тог реда и знак двотачка (:).

Решење 4_1.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX1 80          /*broj znakova u jednom redu*/
#define MAX2 10          /*trazeni broj redova*/

void kopira(int kom, char *ime_dat1, char *ime_dat2);

main(int argc, char *argv[])
{
    switch(argc)          /*argc je broj podataka zadat u komandnoj liniji*/
    {
        case 3:          /*ako su pored imena programa zadata jos dva podatka*/
            /*argv[1] i argv[2] pokazuju na imena datoteka*/
            kopira(0, argv[1], argv[2]);
            break;

        case 4:          /*ako su pored imena programa zadata jos tri podatka*/
            /*ako je argv[1] u traženom obliku*/
            if(strcmp(argv[1], "-num")==0)
                /*argv[2] i argv[3] pokazuju na imena datoteka*/
                kopira(1, argv[2], argv[3]);
            else
            {
                fprintf(stderr, "Pogresna komanda u komandnoj liniji!");
                exit(1);
            }
            break;

        default:
            fprintf(stderr, "Pogresan broj reci u komandnoj liniji!");
            exit(1);
            break;
    }
    return 0;
}

void kopira(int kom, char *ime_dat1, char *ime_dat2)
{
    char red[MAX1+1];
    int i=0, j, l, lmax=0;
    FILE *fptr1, *fptr2;          /*deklaracija pokazivaca za rad sa datotekama*/

    fptr1 = fopen(ime_dat1, "r");          /*otvaranje jedne datoteke */
```



```

if(fp1==NULL)
{
    fprintf(stderr, "Nije uspjelo otvaranje datoteke!");
    exit(1);
}

fp2 = fopen(ime_dat2, "w");    /*otvaranje druge datoteke */
if(fp2==NULL)
{
    fprintf(stderr, "Nije uspjelo otvaranje datoteke!");
    exit(1);
}

while(fgets(red, MAX1+1, fp1) != NULL)/*citanje MAX znakova reda*/
    if(strlen(red)>lmax)
        lmax = strlen(red);        /*nalazenje najveće dužine reda*/

fseek(fp1,0,SEEK_SET);            /*povratak na početak datoteke*/

while(1)
{
    fgets(red, MAX1+1, fp1);i++;
    iffeof(fp1)
        break;

    if(kom==1)                    /*ako je zadana komanda*/
    {
        printf("%3d: ", i);      /*dodavanje rednog broja*/
        fprintf(fp2, "%3d: ", i);
    }
    printf("%s", red);            /*upis procitanog reda*/
    fprintf(fp2, "%s", red);

    if((l==strlen(red))<lmax)      /*ako je red kraći od najdužeg*/
        for(j=0; j<lmax-l;j++)    /*dopuna odgovarajućim znakovima*/
        {
            putchar('^');
            fputc('^',fp2);
        }
}
fcloseall();                    /*zatvaranje svih otvorenih datoteka*/
}

```

Задатак 4_2.

Написати програм на језику C који:

- прихвата из командне линије опцију (*-i* или *-f*) као и низ података,
- по изабраној опцији *-i* формира низ целих бројева од података из командне линије и израчунава средњу вредност овог низа,
- по изабраној опцији *-f* формира низ реалних бројева од података из командне линије и израчунава средњу вредност овог низа,
- уписује елементе низа целих бројева у текст датотеку *celi.txt*, а елементе низа реалних у текст датотеку *realni.txt*.

Решење 4_2.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

float celi(int [],int );
float realni(float [], int );

main(int argc, char *argv[])
{
    int i,n,*niz1;
    float *niz2,srednja;
    FILE *fptr;

    /*argc je broj podataka zadat u komandnoj liniji*/
    if(argc<3) /*ako pored imena programa i opcije nije zadato jos podataka*/
    {
        printf("\nNema dovoljno argumenata \n");
        exit(1);
    }

    /*provera da li je pre niza zadata odgovarajuca opcija*/
    if(strcmp(argv[1],"-i")!=0 && strcmp(argv[1],"-f")!=0)
    {
        printf("\nNepoznata opcija %s\n",argv[1]);
        exit(1);
    }

    n=argc-2; /*u kom. liniji se zadaju: ime programa, opcija i niz*/

    switch(argv[1][1]) /*2. znak iz 2. podatka iz kom. linije je opcija*/
    {
        case 'i':
            /*dodela memorije za niz*/
            niz1=malloc(n*sizeof(int));
            if(niz1==NULL)
            {
                printf("\nNije uspela dodela memorije\n");
                exit(1);
            }

            for(i=0;i<n;i++)
                niz1[i]=atoi(argv[i+2]);

            srednja=celi(niz1,n);
            printf("\nSrednja vrednost niza je: %.2f\n",srednja);

            /*otvaranje datoteke za dodavanje sadrzaja*/
            fptr=fopen("celi.txt","a");
            if(fptr==NULL)
            {
                printf("\n Nije uspelo otvaranje datoteke\n");
                exit(1);
            }
        }
    }
```

```

        /*upis niza iz kom. linije u datoteku*/
        for(i=0;i<n;i++)
            fprintf(fptr,"%d ",niz1[i]);
        fprintf(fptr,"\n");

        /*oslobadjanje memorije i zatvaranje datoteka*/
        free(niz1);
        fclose(fptr);
        break;

    case 'f':
        /*dodela memorije za niz*/
        niz2=malloc(n*sizeof(float));
        if(niz2==NULL)
        {
            printf("\nNije uspela dodela memorije\n");
            exit(1);
        }

        for(i=0;i<n;i++)
            niz2[i]=(float)atof(argv[i+2]);

        srednja=realni(niz2,n);
        printf("\nSrednja vrednost niza je: %.2f\n",srednja);

        /*otvaranje datoteke za dodavanje sadrzaja*/
        fptr=fopen("realni.txt","a");
        if(fptr==NULL)
        {
            printf("\n Nije uspelo otvaranje datoteke\n");
            exit(1);
        }

        /*upis niza iz kom. linije u datoteku*/
        for(i=0;i<n;i++)
            fprintf(fptr,"%f ",niz2[i]);
        fprintf(fptr,"\n");

        /*oslobadjanje memorije i zatvaranje datoteke*/
        free(niz2);
        fclose(fptr);
        break;

    default:
        break;
}

return 0;
}

float celi(int x[],int n)
{
    int i;
    float suma=0;

```

```

        for(i=0;i<n;i++)
            suma+=x[i];

        return suma/n;
}
float realni(float x[], int n)
{
    int i;
    float suma=0;

    for(i=0;i<n;i++)
        suma+=x[i];

    return suma/n;
}

```

Задатак 4_3.

Написати програм на језику C који:

- креира датотеку *redovi.txt* и уписује у ову датотеку редове текста задате преко тастатуре, њих n ($n \leq 20$),
- чита садржај датотеке *redovi.txt* и приказује на екрану редове поређане по дужини (од најкраћег),
- одређује и приказује на екрану средњу дужину свих редова из датотеке *redovi.txt*.

Решење 4_3.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define BROJ 20          /*dozvoljen broj redova*/
#define MAX 80           /*broj znakova u jednom redu*/

FILE *fptr;              /*deklaracija pokazivaca za rad sa datotekom*/

int upis_dat(char *ime);
void sortira_dat(char *ime, int n);

main()
{
    int n;

    /*citanje podataka sa tastature i upis u datoteku*/
    n = upis_dat("redovi.txt");
    /*citanje datoteke, sortiranje redova i stampanje rezultata*/
    sortira_dat("redovi.txt", n);

    return 0;
}

```

```

int upis_dat(char *ime)
{
    int i, n=0;
    char red[MAX+1];

    /*otvaranje datoteke za upis*/
    fptr=fopen(ime,"w");
    if(fptr==NULL)
    {
        printf("Nije uspjelo otvaranje datoteke!");
        exit(1);
    }

    do
    {
        printf("Unesite broj redova n<20: ");
        scanf("%d",&n);
        fflush(stdin); /*citanje zaostalih karaktera sa ulaza*/
    }while(n<1||n>BROJ);

    printf("Unesite %d redova:\n", n);
    for(i=0;i<n;i++)
    {
        gets(red);                                /*citanje reda znakova sa tastature*/

        if(strlen(red)==0)                        /*prazan red znaci kraj unosa*/
            break;

        fputs(red,fptr);                          /*upis reda znakova u datoteku*/
        fputc('\n',fptr);

    }
    n=i;                                          /*n je procitan broj redova*/

    /*zatvaranje datoteke*/
    fclose(fptr);

    return n;
}

void sortira_dat(char *ime, int n)
{
    int i, j;
    float srednja=0;
    char red[MAX+1], redovi[BROJ][MAX+1];

    /*otvaranje datoteke za citanje*/
    if((fptr=fopen(ime,"r"))==NULL)
    {
        printf("Nije uspjelo otvaranje datoteke!");
        exit(1);
    }

    for(i=0;i<n;i++)                            /*citanje n redova znakova iz datoteke*/
        fgets(redovi[i],MAX+1,fptr);

    /*sortiranje redova i stampa rezultata*/
    for(i=0;i<n-1;i++)

```

```

        for(j=i+1;j<n;j++)
            if(strlen(redovi[i])>strlen(redovi[j]))
            {
                strcpy(red,redovi[i]);
                strcpy(redovi[i],redovi[j]);
                strcpy(redovi[j],red);
            }

printf("\nRedovi sortirani po duzini:\n");
for(i=0;i<n;i++)
{
    puts(redovi[i]);
    srednja+=strlen(redovi[i])-1;
}
srednja/=n;                                /*srednja duzina redova*/
printf("\nSrednja duzina redova je %.f.\n", srednja);

/*zatvaranje datoteke*/
fclose(fp);
}

```

Задатак 4_4.

Написати програм на језику C који:

- чита са тастатуре имена од изворне и одредишне текст датотеке,
- копира садржај прве у другу само ако одредишна датотека не постоји (онемогућава случајно преписивање садржаја датотеке) и при томе сва велика слова конвертује у мала,
- приказује на екрану садржај изворне датотеке, по једну реч у сваком реду.

Решење 4_4.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 30                                /*broj znakova u nazivu*/

main()
{
    int c;
    char ime1[MAX+1], ime2[MAX+1];
    FILE *f1, *f2;                            /*deklaracija pokazivaca za rad sa dve datoteke*/

    printf("\nIme izvorne datoteke? ");        gets(ime1);
    printf("\nIme odredisne datoteke? ");      gets(ime2);

    f1 = fopen(ime1,"r");                      /*otvaranje prve datoteke*/
    if(f1==NULL)
    {
        fprintf(stderr," Nije uspeo otvaranje datoteke !");
        exit(1);
    }
}

```

```

    }
    /*da ne bi bila slucajno prepisana proverava se da li datoteka postoji*/
    f2 = fopen(ime2,"r");
    if(f2==NULL)                                /*ako druga datoteka ne postoji*/
    {      f2 = fopen(ime2,"w");                /*otvaranje druge datoteke*/
        if(f2==NULL)
        {      fprintf(stderr," Nije uspeo otvaranje datoteke !");
            exit(1);
        }

        while((c = fgetc(f1)) != EOF)          /*sve do oznake za kraj datoteke*/
        {      if(c==' ' || c=='\t')            /*prikaz njenog sadrzaja rec po rec*/
                putchar('\n');
            else
                putchar(c);
            c = tolower(c);                    /*konverzija velikih u mala slova*/
            fputc(c,f2);                       /*i upis u drugu datoteku*/
        }
    }
    /*zatvaranje otvorenih datoteka*/
    fclose(f1);
    fclose(f2);

    return 0;
}

```

Задатак 4_5.

Написати програм на језику C који:

- чита са тастатуре име од изворне текст датотеке и имена од две одредишне текст датотеке,
- копира у прву одредишну датотеку садржај изворне датотеке тако што избацује сваку реч која почиње цифром или знаком доња црта (_),
- копира у другу одредишну датотеку садржај изворне текст датотеке тако што сваки празан знак замењује знаком доња црта (_), а свако слово после празног знака мења у исто велико слово.

Решење 4_5.

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#define MAX 20                                /*broj znakova u nazivima*/

main()
{      FILE *fptr1,*fptr2,*fptr3;
    char ime1[MAX],ime2[MAX],ime3[MAX];

```

```

char rec[MAX+5],c,cpreth=' ';

printf("\nUnesite ime izvorsne datoteke ");
gets(ime1);
printf("\nUnesite ime prve odredisne datoteke ");
gets(ime2);
printf("\nUnesite ime druge odredisne datoteke ");
gets(ime3);

/*otvaranje datoteka*/
fptr1=fopen(ime1,"r");
if(fptr1==NULL)
{
    printf("\nGreska  otvaranja datoteke \n");
    exit(1);
}
fptr2=fopen(ime2,"w");
if(fptr2==NULL)
{
    printf("\n Nije uspelo otvaranje datoteke \n");
    exit(1);
}
fptr3=fopen(ime3,"w");
if(fptr3==NULL)
{
    printf("\n Nije uspelo otvaranje datoteke \n");
    exit(1);
}

/*kopiranje sadrzaja*/
while(fscanf(fptr1,"%s",rec)!=EOF)      /*dok ima podataka u 1. datoteci*/
{
    if (rec[0]!='_' && (!isdigit(rec[0]))) /*formiranje sadrzaja 2. datoteke*/
        fprintf(fptr2,"%s ",rec);
}

rewind(fptr1);                          /*povratak na pocetak 1. datoteke*/

while((c=fgetc(fptr1))!=EOF)             /*do znaka za kraj 1. datoteke*/
{
    if(c==' ')                          /*formiranje sadrzaja 3. datoteke*/
        fputc('_',fptr3);
    else if(cpreth==' ' || cpreth=='\n')
        fputc(toupper(c),fptr3);
    else
        fputc(c,fptr3);

    cpreth=c;                          /*tekuci znak ce biti prethodni u novoj iteraciji*/
}

/*zatvaranje svih datoteka*/
fcloseall();
return 0;
}

```


Задатак 4_6.

Написати програм на језику C који:

- чита са тастатуре дужину низа n ($n \leq 100$), динамички додељује меморију и чита низ x задате дужине, а онда чита и два цела броја l и k ($l \leq k$),
- формира низ y од низа x тако што вредност сваког елемента низа x која је у опсегу бројева од l до k множи са 10, а на местима елемената који нису у истом опсегу бројева поставља вредност 0,
- понавља све кораке m ($m \leq 10$) пута и додаје елементе низа y сваки пут и текстуалној и бинарној датотеци (име је исто код обе датотеке и задаје се преко тастатуре, а екстензије имена су *.txt* и *.bin*).

Решење 4_6.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BROJ 10          /*dozvoljen broj nizova*/
#define MAX 100         /*dozvoljena duzina niza*/

void cita(int *niz,int n);
void novi(int *niz,int *niz2, int n, int l, int k);
void stampa(int *niz,int n);
void upis(char *ime, int *niz,int n);

main()
{
    int i,n,m,l,k,*x,*y;
    char ime[BROJ*2];

    do
    {
        printf("\nUnesite broj nizova: ");
        scanf("%d",&m);
    }while(m<1 || m>BROJ);
    do
    {
        printf("\nUnesite duzinu niza: ");
        scanf("%d",&n);
    }while(n<1 || n>MAX);
    do
    {
        printf("\nUnesite brojeve l i k takve da je l<=k: ");
        scanf("%d%d",&l,&k);
    }while(l>k);

    printf("\nIme datoteke nizova: ");
    scanf("%s",ime);

    /*dodela memorije za nizove*/
    if((x=malloc(n*sizeof(int)))==NULL)
    {
        printf("\nNije uspela dodela memorije\n");
        exit(1);
    }
}
```

```

        if((y=malloc(n*sizeof(int)))==NULL)
        {
            printf("\nNije uspela dodela memorije\n");
            exit(1);
        }

        for(i=0; i<m; i++)
        {
            cita(x,n);
            novi(x,y,n,l,k);
            stampa(y,n);
            upis(ime, y,n);
        }

        /*oslobadjanje memorije*/
        free(x);
        free(y);

        return 0;
    }
void cita(int niz[],int n)
{
    int i;

    printf("\nUnesite elemente niza: ");
    for(i=0;i<n;i++)
        scanf("%d",&niz[i]);
}
void novi(int niz[],int niz2[], int n, int l, int k)
{
    int i;

    for(i=0;i<n;i++)
        if(niz[i]>=l && niz[i]<=k)
            niz2[i]=niz[i]*10;
        else
            niz2[i]=0;
}
void stampa(int niz[],int n)
{
    int i;

    for(i=0;i<n;i++)
        printf("%d ",niz[i]);
}
void upis(char *ime, int niz[],int n)
{
    int i;
    char ime_dat1[MAX*2],ime_dat2[MAX*2];
    FILE *fptrt, *fptrb;          /*deklaracija pokazivaca za rad sa datotekama*/

    strcpy(ime_dat1,ime);
    strcat(ime_dat1,".txt");
    strcpy(ime_dat2,ime);
    strcat(ime_dat2,".bin");

```

```

/*otvaranje datoteka */
fptrt = fopen(ime_dat1, "a");
if(fptrt==NULL)
{
    fprintf(stderr," Nije uspelo otvaranje datoteke !");
    exit(1);
}
fptrb = fopen(ime_dat2, "ab");
if(fptrb==NULL)
{
    fprintf(stderr," Nije uspelo otvaranje datoteke !");
    exit(1);
}

/*upis u tekst datoteku(za stampu)*/
for(i=0; i<n; i++)
    fprintf(fptrt, "%d ", niz[i]);
fprintf(fptrt, "\n");

/*upis u binarnu datoteku (kraci zapis)*/
fwrite(niz,sizeof(int),n,fptrb);

/*zatvaranje datoteka*/
fcloseall();
}

```

Задатак 4_7.

Написати програм на језику C који:

- чита са тастатуре низ од 10 редова знакова и сортира га по абecedном реду,
- понавља корак под а) 5 пута и сваки пут одређује број замена места редовима при сортирању,
- уписује у текст датотеку за сваки задати низ редова: у посебном реду број замена , а онда и сортирани низ редова.

Решење 4_7.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define BROJ1 5          /*dozvoljen broj nizova*/
#define BROJ2 10        /*dozvoljen broj redova u nizu*/
#define MAX 80          /*broj znakova u redu*/

void cita(char *niz[], int n);
int sortira(char *niz[], int n);
void upis_dat(char *ime, char *niz[], int n, int broj);

```

```

main()
{
    int i, br_zamena;
    char *niz[BROJ2];

    /*dodela memorije za BROJ2 nizova znakova*/
    for(i=0;i<BROJ2;i++)
    {
        niz[i] = malloc(MAX+1);
        if(niz==NULL)
        {
            printf("\nNije uspela dodela memorije\n");
            exit(1);
        }
    }
    for(i=0; i<BROJ1;i++)
    {
        /*citanje niza redova sa tastature*/
        cita(niz, BROJ2);

        /*sortiranje niza redova*/
        br_zamena = sortira(niz, BROJ2);

        /*upis rezultata u datoteku*/
        upis_dat("redovi.txt", niz, BROJ2, br_zamena);
    }
    printf("\nPodaci su upisani u datoteku\n\n");

    /*oslobadjanje memorije*/
    for(i=0;i<BROJ2;i++)
        free(niz[i]);
    return 0;
}

void cita(char *niz[], int n)
{
    int i;
    printf("\nUnesite %d redova teksta\n", n);
    {
        for(i=0;i<n;i++)
            gets(niz[i]);
    }
}

int sortira(char *niz[], int n)
{
    char pom[MAX+1];
    int i,j,br_zamena=0;

    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if(strcmp(niz[i],niz[j])>0)
            {
                strcpy(pom,niz[i]);
                strcpy(niz[i],niz[j]);
                strcpy(niz[j],pom);
                br_zamena++;
            }
    return br_zamena;
}

```

```

void upis_dat(char *ime, char *niz[], int n, int broj)
{
    int i;
    FILE *fptr;          /*deklaracija pokazivaca za rad sa datotekom*/

    /*otvaranje datoteke za dodavanje sadrzaja*/
    fptr=fopen(ime,"a");
    if(fptr==NULL)
    {
        printf("Nije uspeclo otvaranje datoteke!");
        exit(1);
    }

    /*upis broja zamena*/
    fprintf(fptr,"\nBroj zamena: %d\n", broj);
    /*upis sortiranog niza redova*/
    for(i=0;i<n;i++)
    {
        fputs(niz[i],fptr);
        fputc('\n',fptr);
    }

    /*zatvaranje datoteke*/
    fclose(fptr);
}

```

Задатак 4_8.

Написати програм на језику С који:

- динамички додељује меморију за списак од највише 50 станара са следећим подацима за сваког: аутоматски генерисан *redni_broj*, *ime*, *prezime*, *ulica* и *broj_ulaza*,
- чита са тастатуре податке за n ($n \leq 50$) станара и формира од ових података записе у бинарној датотеци *adrese.bin*,
- чита из датотеке *adrese.bin* и приказује на екрану редни број, име и презиме за све станаре на задатој адреси: *ulica* и *broj_ulaza* (ако задата адреса не постоји у списку пријављује на екрану одговарајућу поруку).

Решење 4_8.

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define BROJ 50          /*dozvoljen broj stanara*/
#define MAX 80           /*broj znakova u nazivima*/

typedef struct{
    int rbr,br_ulaza;
    char ime[MAX+1],ulica[MAX+1];
}Stanari;

```

```

FILE *fptr;

int cita(Stanari niz[]);
void upis(Stanari niz[], int n);
int pretraga(void);

main()
{
    Stanari *niz;
    int n,nadjen;

    /*dodela memorije za niz struktura*/
    niz=malloc(BROJ*sizeof(Stanari));
    if(niz==NULL)
    {
        printf("\nNije uspela dodela memorije\n");
        exit(1);
    }

    n=cita(niz);

    /*promena velicine dodeljene memorije*/
    niz=realloc(niz,n*sizeof(Stanari));
    if(niz==NULL)
    {
        printf("\nNije uspela dodela memorije\n");
        exit(1);
    }

    upis(niz, n);
    nadjen=pretraga();
    if (nadjen==0)
        printf("\nNema stanara na toj adresi\n");
    printf("\n\n");

    return 0;
}

int cita(Stanari x[])
{
    int i;

    for(i=0;i<BROJ;i++)
    {
        printf("\nIme i prezime stanara (za kraj: stop): ");
        gets(x[i].ime);
        if(strcmp(x[i].ime,"stop")==0)
            break;

        printf("\nUlica: ");
        gets(x[i].ulica);
        printf("\nBroj ulaza: ");
        scanf("%d",&x[i].br_ulaza);
        x[i].rbr=i+1;
        fflush(stdin); /*citanje zaostalih karaktera sa ulaza*/
    }
}

```

```

        return i;          /*f-ja vraca broj stanara*/
    }
void upis(Stanari niz[], int n)
{
    /*otvaranje binarne datoteke */
    fptr=fopen("adrese.bin","wb");
    if(fptr==NULL)
    {
        printf("\nGreska kod otvaranja datoteke\n");
        exit(1);
    }

    /*upis niza struktura*/
    fwrite(niz,sizeof(Stanari),n,fptr);

    /*zatvaranje datoteke*/
    fclose(fptr);
}
int pretraga(void)
{
    int nadjen=0,broj;
    char ulica[MAX+1];
    Stanari s;

    /*citanje podataka za pretragu*/
    printf("\nUlica za pretragu:");
    gets(ulica);
    printf("\nBroj ulaza: za pretragu");
    scanf("%d",&broj);
    fflush(stdin);          /*citanje zaostalih karaktera sa ulaza*/

    /*otvaranje datoteke*/
    fptr=fopen("adrese.bin","rb");
    if(fptr==NULL)
    {
        printf("\nG Nije uspelo otvaranje datoteke\n");
        exit(1);
    }

    /*prikaz spiska*/
    printf("\nNa adresi %s %d stanuju: \n\n",ulica,broj);
    while(1)
    {
        fread(&s,sizeof(Stanari),1,fptr);

        if(feof(fptr))
            break;

        if(s.br_ulaza==broj && strcmp(s.ulica,ulica)==0)
        {
            nadjen=1;
            printf("\n%d. %s ",s.rbr,s.ime);
        }
    }
}

```

```

        /*zatvaranje datoteke*/
        fclose(fptr);

        return nadjen;                /*f-ja vraca status da li su nadjeni podaci*/
    }

```

Задатак 4_9.

Написати програм на језику C који:

- чита са тастатуре податке за n ($n \leq 200$) производа (*sifra*, *naziv*, *cena*) и од њих и аутоматски генерисаног податка *redni_broj* формира записе у бинарној датотеци *proizvodi.bin*,
- од свих записа редом прочитаних из датотеке *proizvodi.bin*. издваја оне код којих је *cena* у опсегу вредности задатих преко тастатуре и формира од њих записе у текстуалној датотеци *izbor.txt*,
- приказује на екрану садржај текстуалне датотеке *izbor.txt*,

Решење 4_9.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX2 200                /*dozvoljen broj proizvoda*/
#define MAX1 20                 /*broj znakova u nazivima*/

struct proizvod{                /*definicija tipa strukture*/
    int redni_broj;
    int sifra;
    char naziv[MAX1+1];
    int cena;
};

void formira(FILE *fptr1);
void izbor(FILE *fptr1, FILE *fptr2);
void prikaz(FILE *fptr2);

int broj_zapisa_b;               /*broj zapisa u binarnoj datoteci*/
int broj_zapisa_t;              /*broj zapisa u tekst datoteci*/

main()
{
    FILE *fptr1, *fptr2;        /*deklaracija pokazivaca za rad sa datotekama*/

    /*otvaranje datoteka*/
    fptr1 = fopen("proizvod.bin", "wb+");
    if(fptr1==NULL)
    {
        fprintf(stderr, "Nije uspjelo otvaranje datoteke");
        exit(1);
    }
}

```



```

    }

    fptr2 = fopen("izbor.txt", "w+");
    if(fptr2==NULL)
    {
        fprintf(stderr,"Nije uspelo otvaranje datoteke");
        exit(1);
    }

    formira(fptr1);
    izbor(fptr1,fptr2);
    prikaz(fptr2);

    /*zatvaranje datoteka*/
    fclose(fptr1);
    fclose(fptr2);

    return 0;
}

void formira(FILE *fptr1)
{
    int i;
    struct proizvod zapis;

    do
    {
        printf("\nUneti broj proizvoda: ");
        scanf("%d", &broj_zapisa_b);
    }while(broj_zapisa_b<1 || broj_zapisa_b>MAX2);

    printf("\nUneti za svaki proizvod, u posebnoj liniji");
    printf("\nsifru, naziv i cenu:\n");

    for(i=0; i<broj_zapisa_b; i++)
    {
        zapis.redni_broj = i+1;

        scanf("%d%s%d",&zapis.sifra,zapis.naziv,&zapis.cena);
        fwrite(&zapis,sizeof(struct proizvod),1,fptr1);
    }
}

void izbor(FILE *fptr1, FILE *fptr2)
{
    int i, min, max;
    struct proizvod zapis;

    printf("\nUneti opseg cena: ");
    scanf("%d%d", &min, &max);

    fseek(fptr1,0,SEEK_SET);

    broj_zapisa_t=0;
    for(i=0; i<broj_zapisa_b; i++)
    {
        fread(&zapis,sizeof(struct proizvod),1,fptr1);
    }
}

```

```

        if(zapis.cena>=min && zapis.cena<=max)
        {
            broj_zapisa_t++;
            fprintf(fp2, "%d %d %s %d\n",zapis.redni_broj,
                zapis.sifra,zapis.naziv,zapis.cena);}
    }
}
void prikaz(FILE *fp2)
{
    int i;
    struct proizvod zapis;

    fseek(fp2,0,SEEK_SET);

    printf("\nSpisak proizvoda u zadatom opsegu cena:\n");
    for(i=0; i<broj_zapisa_t; i++)
    {
        fscanf(fp2, "%d%d%s%d",&zapis.redni_broj,
            &zapis.sifra,zapis.naziv,&zapis.cena);
        printf("%d. %d %s %d\n",  zapis.redni_broj,
            zapis.sifra,zapis.naziv,zapis.cena);
    }
}

```

Задатак 4_10.

Написати програм на језику C који:

- креира списак и попуњава га подацима задатим преко тастатуре за n ($n \leq 10$) књига: *ime_knjige*, *ime_autora*, *god_izdanja*, *tiraz* и *prodatih*,
- за сваку књигу из списка уписује у бинарну датотеку *knjige.bin*: редни број и све задате податке,
- приказује на екрану податке о свакој књизи из датотеке *knjige.bin*, а по задатом редном броју књиге чита са тастатуре и ажурира у истој датотеци податак за изабрану књигу: *prodatih*.

Решење 4_10.

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 10                /*dozvoljen broj knjiga*/
#define MAX2 30               /*broj znakova u nazivima*/

struct knjiga{                /*definicija tipa strukture*/
    char ime[MAX2+1];
    char autor[MAX2+1];
    int god, tiraz, prodatih;
};

void cita(struct knjiga niz[], int n);
void azurira(char *ime, int n);
void stampa(char *ime, int n);

```

```

main()
{
    int n;
    struct knjiga *niz;          /*deklaracija pokazivaca na niz struktura*/

    do
    {
        printf("\nBroj knjiga? ");
        scanf("%d", &n);          /*posle poziva funkcije scanf*/
        fflush(stdin);            /*citanje zaostalih karaktera sa ulaza*/
    }while(n<1 || n>MAX);

    /*dodela memorije*/
    niz = malloc(n*sizeof(struct knjiga));
    if(niz==NULL)
    {
        fprintf(stderr, "Nije uspela dodela memorije!");
        exit(1);
    }

    cita(niz, n);
    stampa("knjige.bin", n);
    azurira("knjige.bin", n);
    stampa("knjige.bin", n);

    /*oslobadjanje memorije*/
    free(niz);

    return 0;
}

void cita(struct knjiga niz[], int n)
{
    FILE *f;                      /*deklaracija pokazivaca za rad sa datotekom*/
    int i;

    printf("\nPodaci o knjigama\n(ime i autor, svaki u posebnom redu)");
    printf("\n(godina izdanja, tiraz i prodatih u trecem redu)? ");
    for(i=0; i<n; i++)
    {
        printf("\n\n%d. knjiga:\n", i+1);
        gets(niz[i].ime);
        gets(niz[i].autor);
        scanf("%d%d%d", &niz[i].god, &niz[i].tiraz, &niz[i].prodatih);
        fflush(stdin);            /*citanje zaostalih karaktera sa ulaza*/
    }

    f = fopen("knjige.bin", "wb"); /*otvaranje binarne datoteke */
    if(f==NULL)
    {
        fprintf(stderr, " Nije uspelo otvaranje datoteke !");
        exit(1);
    }
    fwrite(niz, sizeof(struct knjiga), n, f); /*upis n struktura u datoteku*/
    fclose(f);                             /*zatvaranje datoteke*/
}

void azurira(char *ime, int n)
{
    struct knjiga k1;
    FILE *f;                              /*deklaracija pokazivaca za rad sa datotekom*/

```

```

int rbr, prodatih;
while(1)
{
    printf("\nRedni broj knjige?");
    printf("\n(za kraj zadati redni broj van postojećih): ");
    scanf("%d", &rbr);
    if(rbr<1 || rbr>n)
        break;

    f = fopen(ime,"rb+"); /*otvaranje binarne datoteke */
    if(f==NULL)
    {
        fprintf(stderr," Nije uspjelo otvaranje datoteke!");
        exit(1);
    }

    /*pomeraj u datoteci od pocetka za (rbr-1) struktura*/
    fseek(f,(rbr-1)*sizeof(struct knjiga),SEEK_SET);
    /*citanje jedne strukture podataka*/
    fread(&k1,sizeof(struct knjiga),1,f);

    printf("Prodatih? ");
    scanf("%d", &prodatih);
    k1.prodatih += prodatih;          /*azuriranje broja prodatih*/

    /*pomeraj u datoteci od tekuće pozicije za jednu strukturu unazad*/
    fseek(f,-(long)sizeof(struct knjiga),SEEK_CUR);
    /*upis u datoteku jedne strukture podataka*/
    fwrite(&k1,sizeof(struct knjiga),1,f);

    fclose(f);                      /*zatvaranje datoteke*/
}
}

void stampa(char *ime, int n)
{
    int i;
    struct knjiga k1;
    FILE *f;                        /*deklaracija pokazivaca za rad sa datotekom*/

    f = fopen(ime,"rb");            /*otvaranje binarne datoteke */
    if(f==NULL)
    {
        fprintf(stderr, " Nije uspjelo otvaranje datoteke!");
        exit(1);
    }
    /*citanje n struktura podataka i prikaz na ekranu*/
    printf("\nSpisak:\n");
    for(i=0; i<n; i++)
    {
        fread(&k1,sizeof(struct knjiga),1,f);
        puts(k1.ime);
        puts(k1.autor);
        printf("%d %d %d\n\n", k1.god, k1.tiraz, k1.prodatih);
    }
    fclose(f);                      /*zatvaranje datoteke*/
}

```

**5. Оператори, наредбе,
функције, низови, структуре,
динамичка додела меморије,
аргументи из командне линије
и рад са датотекама
у програмима на језику C++**

Задатак 5_1.

Написати програм на језику C++ који:

- а) прихвата из командне линије речи: `-o ime1 ime2` (где је `o` опција у облику једног слова, `ime1` и `ime2` имена изворне и одредишне текст датотеке),
- б) уписује на почетак одредишне датотеке коментар: `/*Ovo je kopija datoteke*/`
- в) копира садржај изворне датотеке у одредишну датотеку на следећи начин: по добијеној опцији `-t` из командне линије управљачки знак `'\t'` чини видљивим (представља га словима: `"HT"`), а по изабраној опцији `-b` замењује сваки управљачки знак `'\t'` једним празним знаком.

Решење 5_1.

```
#include <fstream.h>
#include <stdlib.h>
#include <string.h>

main(int argc, char *argv[])
{
    int c;
    fstream f1, f2;           //kreiranje objekata za rad sa datotekama
    if(argc<4)                //ako pored imena programa nisu zadata jos 3 podatka
    {
        cerr<<"Nedovoljan broj parametara!";
        exit(1);
    }

    //otvaranje jedne tekst datoteke za citanje
    f1.open(argv[2],ios::in);  //argv[2] pokazuje na string koji cuva ime
    if(!f1)
    {cerr<<" Nije uspeclo otvaranje datoteke!"<<endl;
      exit(1);
    }

    //otvaranje druge tekst datoteke
    f2.open(argv[3],ios::out); //argv[3] pokazuje na string koji cuva ime
    if(!f2)
    {
        cerr<<" Nije uspeclo otvaranje datoteke!"<<endl;
        exit(1);
    }

    //upis teksta zaglavlja u datoteku
    f2<<"/*Ovo je kopija datoteke*/"<<endl;

    //kopiranje sadrzaja jedne u drugu datoteku na odgovarajuci nacin
    if(strcmp(argv[1],"-t")==0) //argv[1] pokazuje na string koji cuva opciju
    {
        while((c = f1.get()) != EOF)
        {
            if(c=='\t')                //zamena znaka '\t' slovima HT
                f2<<"HT";
            else
                f2.put((char)c);
        }
    }
```

```

else if(strcmp(argv[1], "-b")==0)
{
    while((c = f1.get()) != EOF)
        if(c=='\t') //zamena znaka '\t' praznim znakom
            f2<<' ';
        else
            f2.put((char)c);
}
else
    cout<<"Izabrana pogresna opcija";

//zatvaranje datoteka
f1.close();
f2.close();

return 0;
}

```

Задатак 5_2.

Написати програм на језику C++ који:

- прихвата из командне линије имена од три текст датотеке,
- кодиран садржај прве датотеке (дуплирани знакови <, >, @) копира у другу датотеку,
- декодиран садржај друге датотеке (замењени дуплирани знакови <, > и @ истим једноструким знаковима) копира у трећу датотеку.

Решење 5_2.

```

#include <fstream.h>
#include<stdlib.h>

void kodira(char *ime1, char *ime2);
void dekodira(char *ime1, char *ime2);

main(int argc,char *argv[])
{
    if(argc<4) //ako pored imena programa nisu zadata jos 3 podatka
    {
        cerr<<"Nedovoljan broj parametara!";
        exit(1);
    }

    kodira(argv[1], argv[2]);
    dekodira(argv[2], argv[3]);

    return 0;
}

void kodira(char *ime1, char *ime2)
{
    int c;
    fstream f1,f2; //kreiranje objekata za rad sa datotekama

```

```

//otvaranje jedne tekst datoteke za citanje
f1.open(ime1, ios::in | ios::nocreate);
if(!f1)
{
    cerr << "\n Nije uspjelo otvaranje datoteke" << endl;
    cerr << "Datoteka sa zadatim imenom ne postoji!" << endl;
    exit(1);
}

//otvaranje druge tekst datoteke
f2.open(ime2, ios::out);
if(!f2)
{
    cerr << "\n Nije uspjelo otvaranje datoteke" << endl;
    cerr << "Datoteka sa zadatim imenom vec postoji!" << endl;
    exit(1);
}

//upis kodiranog sadrzaja prve datoteke u drugu
while((c = f1.get()) != EOF)
{
    if(c=='<'||c=='>'||c=='@')
        f2.put((char)c);
    f2.put((char)c);
}

//zatvaranje datoteka
f1.close();
f2.close();
}

void dekodira(char *ime1, char *ime2)
{
    int c,parni=0;
    fstream f1,f2;          //kreiranje objekata za rad sa datotekama

    //otvaranje jedne tekst datoteke za citanje
    f1.open(ime1, ios::in | ios::nocreate);
    if(!f1)
    {
        cerr << "\n Nije uspjelo otvaranje datoteke " << endl;
        cerr << "Datoteka sa zadatim imenom ne postoji!" << endl;
        exit(1);
    }

    //otvaranje druge tekst datoteke za upis
    f2.open(ime2, ios::out);
    if(!f2)
    {
        cerr << "\n Nije uspjelo otvaranje datoteke " << endl;
        cerr << "Datoteka sa zadatim imenom vec postoji!" << endl;
        exit(1);
    }

    while((c = f1.get()) != EOF)
    {
        if(c=='<'||c=='>'||c=='@')

```



```

        {
            parni++;
            if(parni==1)
                continue;
            parni=0;
        }
        f2.put((char)c);
    }
    //zatvaranje datoteka
    f1.close(); f2.close();
}

```

Задатак 5_3.

Написати програм на језику C++ који:

- прихвата из командне линије имена од две текстуалне датотеке,
- приказује на екрану редове од n -тог до m -тог из прве датотеке (n и m су вредности задате преко тастатуре) или приказује одговарајућу поруку ако датотека нема овај број редова,
- копира садржај прве датотеке у другу тако да сви редови текста буду исте дужине као најкраћи ред у овом тексту (одбацује вишак последњих знакова).

Решење 5_3.

```

#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#define MAX 80                //broj znakova u jednom redu

void prikaz(char *ime_dat, int n, int m);
int min_red(char *ime_dat1);
void kopira(char *ime_dat1, char *ime_dat2, int lmin);

main(int argc, char *argv[])
{
    int n, m, lmin;

    cout<<"Uneti vrednosti n i m: ";
    cin>>n>>m;

    if(argc<3)
    {
        cerr << "Pogresan broj reci u komandnoj liniji!";
        exit(1);
    }
    prikaz(argv[1], n, m);
    lmin = min_red(argv[1]);
    kopira(argv[1], argv[2], lmin);

    return 0;
}

```

```

void prikaz(char *ime_dat, int n, int m)
{
    int i=0;
    char red[MAX+1];
    fstream f;           //kreiranje objekta za rad sa datotekom

    //otvaranje tekst datoteke za citanje
    f.open(ime_dat, ios::in);
    if(!f)
    {
        cerr<<"Nije uspelo otvaranje datoteke!";
        exit(1);
    }

    //prikaz redava od n-tog do m-tog
    while(!f.eof())
    {
        f.getline(red, MAX+1);
        i++;
        if(i>=n && i<=m)
            cout << red << endl;
    }
    if(i<m)
        cout <<"\nKraj datoteke pre " << i+1 << ". reda" << endl;

    //zatvaranje datoteke
    f.close();
}

int min_red(char *ime_dat1)
{
    char red[MAX+1];
    int lmin;

    fstream f1;           //kreiranje objekta za rad sa datotekom

    //otvaranje tekst datoteke za citanje
    f1.open(ime_dat1, ios::in);
    if(!f1)
    {
        cerr<<"Nije uspelo otvaranje datoteke!";
        exit(1);
    }

    //nalazenje najkraceg reda
    f1.getline(red, MAX+1);
    lmin = strlen(red);

    while(!f1.eof())
    {
        f1.getline(red, MAX+1);
        if(strlen(red)<lmin && strlen(red)!=0)
            lmin = strlen(red);
    }
    //zatvaranje datoteke
    f1.close();
    return lmin;
}

```

```

void kopira(char *ime_dat1, char *ime_dat2, int lmin)
{
    char red[MAX+1], bafer[MAX+1];

    fstream f1, f2;           //kreiranje objekata za rad sa datotekama

    //otvaranje tekst datoteka (prve za citanje a druge za upis)
    f1.open(ime_dat1, ios::in);
    if(!f1)
    {
        cerr<<"Nije uspjelo otvaranje datoteke!";
        exit(1);
    }
    f2.open(ime_dat2, ios::out);
    if(!f2)
    {
        cerr<<"Nije uspjelo otvaranje datoteke!";
        exit(1);
    }

    //upis u drugu datoteku lmin znakova od svakog reda prve
    while(!f1.eof())
    {
        f1.get(red, lmin+1);
        f1.getline(bafer, MAX+1);

        f2 << red << endl;
    }

    //zatvaranje datoteka
    f1.close();
    f2.close();
}

```

Задатак 5_4.

Написати програм на језику C++ који:

- прихвата из командне линије дужину низа n ($n \leq 10$), као и саме вредности елемената низа,
- при сваком свом извршавању уписује вредности свих елемената низа, задате у командној линији, у бинарну датотеку *nizovi.bin* (у продужетку постојећег садржаја) и приказује на екрану садржај датотеке *nizovi.bin*,
- уписује у бинарну датотеку *nizovi2.bin* све низове из датотеке *nizovi.bin* чија је дужина у опсегу од *min* до *max* (*min* и *max* су бројеви задати преко тастатуре) и приказује на екрану садржај датотеке *nizovi2.bin*.

Решење 5_4.

```

#include <fstream.h>
#include <stdlib.h>
#define MAX 10           //dozvoljen broj elemenata niza

```

```

void upis(char *ime, int niz[], int n);
void kopira(char *ime1, char *ime2, int niz[]);
void stampa(char *ime, int niz[]);

main(int argc, char *argv[])
{
    int i,n,*niz;

    if(argc<3)           //ako pored imena programa nisu zadata jos 2 podatka
    {
        cerr<<"Nedovoljan broj parametara!";
        exit(1);
    }

    n = atoi(argv[1]);    //argv[1] pokazuje na string koji cuva duzinu niza
    if(n>0 && n<=10)     //oslobadjanje memorije
    {
        //dodela memorije za niz
        niz=new int[n];
        if(niz==NULL)
        {
            cerr<<"Nije uspela dodela memorije!!!"<<endl;
            exit(1);
        }

        for(i=0; i<n; i++)           //argv[2]...argv[n+1] pokazuju na
            niz[i] = atoi(argv[i+2]); //stringove koji cuvaju elemente niza

        upis("nizovi.bin",niz,n);

        cout<<"\nSadrzaj 1. datoteke"<<endl;
        stampa("nizovi.bin", niz);

        kopira("nizovi.bin", "nizovi2.bin", niz);

        cout<<"\nSadrzaj 2. datoteke"<<endl;
        stampa("nizovi2.bin", niz);

        //oslobadjanje memorije
        delete niz;
    }
    return 0;
}

void upis(char *ime, int niz[], int n)
{
    int m=n;
    fstream f;           //kreiranje objekta za rad sa datotekom

    //otvaranje binarne datoteke za dodavanje sadržaja
    f.open("nizovi.bin",ios::app | ios::binary);
    if(!f)
    {
        cerr<<" Nije uspelo otvaranje datoteke !"<<endl;
        exit(1);
    }
}

```

```

        //upis u datoteku duzine niza n i samih n elemenata
        f.write((char *)&m,sizeof(int));
        f.write((char *)niz,n*sizeof(int));

        //zatvaranje datoteke
        f.close();
    }
void kopira(char *ime1, char *ime2, int niz[])
{
    int n, min,max;
    fstream f, f2;                //kreiranje objekata za rad sa datotekama

    do
    {
        cout<<"\nmin i max?";
        cin>>min>>max;
    }while(max<0 && max>10);

    //otvaranje prve datoteke za citanje, druge za upis
    f.open(ime1,ios::in|ios::binary);
    if(!f)
    {
        cerr<<" Nije uspjelo otvaranje datoteke !"<<endl;
        exit(1);
    }
    f2.open(ime2,ios::out|ios::binary);
    if(!f2)
    {
        cerr<<" Nije uspjelo otvaranje datoteke !"<<endl;
        exit(1);
    }

    //izdvajanje nizova cija je duzina u trazenom opsegu
    while(f.read((char *)&n,sizeof(int))!=NULL)
    {
        f.read((char *)niz,n*sizeof(int));
        if(n>=min && n<=max)
        {
            f2.write((char *)&n,sizeof(int));
            f2.write((char *)niz,n*sizeof(int));
        }
    }
    //zatvaranje datoteka
    f.close();
    f2.close();
}
void stampa(char *ime, int niz[])
{
    int i=0, j, n;
    fstream f;                //kreiranje objekta za rad sa datotekom

    //otvaranje binarne datoteke za citanje
    f.open(ime,ios::in|ios::binary);
    if(!f)
    {
        cerr<<" Nije uspjelo otvaranje datoteke !"<<endl;
        exit(1);
    }
}

```

```

//citanje sadrzaja datoteke
while(f.read((char *)&n,sizeof(int))!=NULL)
{
    cout<<"Niz "<<i+1<<": ";i++;
    f.read((char *)niz,n*sizeof(int));
    for(j=0;j<n;j++)
        cout<<niz[j]<<' ';
    cout<<endl;
}

//zatvaranje datoteke
f.close();
}

```

Задатак 5_5.

Написати програм на језику C++ који:

- прихвата из командне линије број низова m ($m \leq 10$) а онда m пута дужину сваког низа n ($n \leq 10$), као и саме вредности елемената низа,
- сваки низ који има нула елементе уписује у бинарну датотеку *nizovi1.bin* а остале низове у бинарну датотеку *nizovi2.bin*,
- у зависности од изабране опције преко тастатуре приказује на екрану садржај датотеке *nizovi1.bin* или *nizovi2.bin*.

Решење 5_5.

```

#include <fstream.h>
#include <stdlib.h>
#define MAX 10          //dozvoljena duzina niza

void cita(int,int);
void meni(int n);
void stampa(char *,int);

main()
{
    int m,n;

    cout<<"\nBroj nizova i duzina svakog niza: ";
    while(1)
    {
        cin>>m>>n;
        if(m>0 && m<=MAX && n>0 && n<=MAX)
            break;
        cout<<"\nBroj nizova i duzina svakog u opsegu od 1 do "<<MAX<<": ";
    }
    cita(m,n);
    meni(n);

    return 0;
}

```

```

void cita(int m,int n)
{
    int *niz;
    fstream f1,f2;          //kreiranje objekata za rad sa datotekama

    //otvaranje datoteka
    f1.open("primer1.bin",ios::out);
    if(!f1)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }
    f2.open("primer2.bin",ios::out);
    if(!f2)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }

    //dodela memorije
    niz= new int [n];
    if(niz==NULL)
    {
        cout<<"\nNije uspela dodela memorije\n";
        exit(1);
    }

    //citanje nizova sa tastature i upis u datoteke
    for(int i=0;i<m;i++)
    {
        int nadjen=0;

        cout<<"\nElementi "<<i+1<<" niza: ";
        for(int j=0;j<n;j++)
        {
            cin>>niz[j];
            if(niz[j]==0)          //ako niz ima nula elemenata
                nadjen=1;        //postavljanje statusa nadjen
        }

        if(!nadjen)                //ako nisu nadjeni nula elementi
            f1.write((char*)niz,n*sizeof(int));
        else                       //u suprotnom
            f2.write((char*)niz,n*sizeof(int));
    }

    //oslobadjanje memorije i zatvaranje datoteka
    delete [] niz;
    f1.close();f2.close();
}

void meni(int n)
{
    int opcija, izlaz=0;

    while(1)
    {
        cout<<"\nPregled datoteka: "<<endl;
        cout<<"\n1. Nizovi bez nula elemenata: ";
    }
}

```

```

        cout<<"\n2. Nizovi sa nula elementima: ";
        cout<<"\n\nVas izbor :";
        cin>>opcija;

        switch(opcija)
        {
            case 1:
                stampa("primer1.bin",n);
                break;
            case 2:
                stampa("primer2.bin",n);
                break;
            default:
                cout<<"\nNepoznata opcija\n";
                izlaz=1;
                break;
        }
        if(izlaz)
            break;
    }
}

void stampa(char *ime,int n)
{
    int *niz,ima=0;
    fstream f;           //kreiranje objekta za rad sa datotekom

    //otvaranje datoteke
    f.open(ime,ios::in|ios::nocreate);
    if(!f)
    {
        cout<<"\ngreska kod otvaranja datoteke\n";
        exit(1);
    }

    //dodela memorije
    niz=new int[n];
    if (niz==NULL)
    {
        cout<<"\nNije uspela dodela memorije\n";
        exit(1);
    }

    //citanje trazenog sadrzaja datoteke
    cout<<"\nSadrzaj trazene datoteke je:\n";
    while(1)
    {
        f.read((char*)niz,n*sizeof(int));
        if(f.eof())
            break;
        ima++;
        for(int i=0;i<n;i++)
            cout<<*(niz+i)<<" ";
        cout<<endl;
    }
}

```



```

        if(!ima)
            cout<<"\nDatoteka je prazna\n";

        //oslobadjanje memorije i zatvaranje datoteke
        delete [] niz;
        f.close();
    }

```

Задатак 5_6.

Написати програм на језику C++ који:

- креира датотеку *reci.txt* и уписује у ову датотеку речи задате преко тастатуре (реч је низ знакова између два бела знака), њих n ($n \leq 100$) или до речи *kraj*,
- чита садржај датотеке *reci.txt* и приказује на екрану сваку прочитану реч у посебном реду,
- одређује и приказује на екрану средњу дужину свих речи из датотеке *reci.txt*.

Решење 5_6.

```

#include <fstream.h>
#include <stdlib.h>
#include <string.h>
#define BROJ 100           //dozvoljen broj reci
#define MAX 80             //broj znakova u nazivu

fstream f;                //kreiranje objekta za rad sa datotekom

void cita(char *ime);
void stampa(char *ime);

main()
{
    char ime[MAX+1];

    cout<<"Zadati ime datoteke: ";
    cin>>ime;

    cita(ime);
    stampa(ime);

    return 0;
}

void cita(char *ime)
{
    int i;
    char rec[MAX+1];

```

```

//otvaranje datoteke
f.open(ime,ios::out);
if(!f)
{
    cerr<<" Nije uspeo otvaranje datoteke !"<<endl;
    exit(1);
}

cout<<"Zadati same reci (do reci \"kraj\"): "<<endl;
for(i=0;i<BROJ;i++)
{
    cin>>rec;

    if(strcmp("kraj",rec)==0)
        break;

    f<<rec<<" ";
    f.put("\n");
}

//zatvaranje datoteke
f.close();
}

void stampa(char *ime)
{
    int n=0;
    float srednja=0;
    char rec[MAX+1];

    //otvaranje datoteke za citanje
    f.open(ime,ios::in);
    if(!f)
    {
        cerr<<" Nije uspeo otvaranje datoteke !"<<endl;
        exit(1);
    }

    cout<<"\nSadrzaj datoteke: "<<endl;
    while(!f.eof())
    {
        f.getline(rec,MAX);
        srednja+=strlen(rec);
        n++;
        cout<<rec<<endl;
    }
    srednja/=n;
    cout<<"srednja duzina reci je: "<<srednja<<endl;

    //zatvaranje datoteke
    f.close();
}

```

Задатак 5_7.

Написати програм на језику C++ који:

- а) чита са тастатуре имена од изворне текст датотеке и од две одредишне текст датотеке,
- б) под условом да прва одредишна датотека не постоји копира у њу садржај изворне датотеке тако што избацује сваки ред који почиње цифром,
- в) под условом да друга одредишна датотека не постоји копира у њу садржај изворне текст датотеке тако што низ од више узастопних празних знакова замењује једним празним знаком.

Решење 5_7.

```
#include<fstream.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#define MAX 20          //broj znakova u nazivima

ifstream fin;           //kreiranje objekta za citanje datoteke
ofstream fout1,fout2;   //kreiranje objekata za upis u datoteke

void kreira_prvu(void);
void kreira_drugu(void);

main()
{
    char ime[MAX+1],ime1[MAX+1],ime2[MAX+1];

    //zadavanje imena datoteka
    cout<<"\nUnesite ime izvorne tekst datoteke: ";
    cin.getline(ime,MAX+1);

    cout<<"\nUnesite ime prve odredisne tekst datoteke ";
    cin.getline(ime1,MAX+1);

    cout<<"\nUnesite ime druge odredisne tekst datoteke ";
    cin.getline(ime2,MAX+1);

    //otvaranje prve datoteke za citanje
    fin.open(ime,ios::nocreate);
    if(!fin)
    {
        cout<<"\nIzvorisna datoteka ne postoji\n";
        exit(1);
    }

    //otvaranje prve odredisne datoteke za upis
    fout1.open(ime1,ios::noreplace);
    if(!fout1)
        cout<<"\nPrva odredisna datoteka vec postoji\n";
```

```

else
    kreira_prvu();

//zatvaranje datoteka
fin.close();
fout1.close();

//otvaranje izvorne datoteke za citanje
fin.open(ime);
if(!fin)
{
    cout<<"\nGreska otvaranja datoteke za citanje \n";
    exit(1);
}

//otvaranje druge odredisne datoteke za upis
fout2.open(ime2,ios::noreplace);
if(!fout2)
    cout<<"\nDruga odredisna datoteka vec postoji\n";
else
    kreira_drugu();

//zatvaranje datoteka
fin.close();
fout2.close();

return 0;
}

void kreira_prvu(void)
{
    char red[MAX+1];
    while(1)
    {
        fin.getline(red,MAX+1);

        if(fin.eof())    //ako je nadjen kraj datoteke
            break;      //kraj petlje

        if(!(isdigit(red[0])))
            fout1<<red<<endl;
    }
}

void kreira_drugu(void)
{
    char c,prethc='\0';

    while((c=fin.get())!=EOF)    //EOF je znak za kraj petlje
    {
        if(c==' ' && prethc==' ')
            continue;
        else
            fout2.put(c);
        prethc=c;
    }
}

```

Задатак 5_8.

Написати програм на језику C++ који:

- а) од m ($m \leq 10$) низова реалних бројева (сваки дужине n ($n \leq 10$)) задатих преко тастатуре формира записе у бинарној датотеци,
- б) записе формиране датотеке разврстава у три бинарне датотеке *poz.bin*, *neg.bin* и *nula.bin* тако да у првој буду записи (низови) у којима има више позитивних елемената него осталих, у другој више негативних него осталих а у трећој преостали записи,
- в) приказује на екрану садржај једне од датотека формираних под б), у зависности од опције (1 / 2 / 3) изабране са тастатуре.

Решење 5_8.

```
#include <fstream.h>
#include <stdlib.h>
#define BROJ 10 //dozvoljen broj nizova
#define MAX 20 //broj znakova u nazivima

fstream f,f1,f2,f3; //kreiranje objekata za rad sa datotekom
int bpoz=0, bneg=0, bnula=0; //broj poz., neg., i nula nizova

void upis(char *ime, int n, int m);
void razvrst(char *ime, int n, int m);
void meni(int n);
void prikaz(char *ime,int broj,int n);

main()
{
    int n,m;
    char ime[MAX+1];

    do
    {
        cout<<"Unesite broj nizova (m<10): ";
        cin>>m;
    }while(m<1||m>BROJ);

    do
    {
        cout<<"Unesite broj elemenata niza (n <10): ";
        cin>>n;
    }while(n<1||n>BROJ);

    cout<<"\nIme datoteke: ";
    cin>>ime;

    upis(ime, n, m);
    razvrst(ime, n,m);
    meni(n);

    return 0;
}
```

```

void upis(char *ime, int m, int n)
{
    int i,j;
    float niz[MAX];
    fstream f;                                //kreiranje objekta za rad sa datotekom

    //otvaranje binarne datoteke
    f.open(ime,ios::out|ios::binary);
    if(!f)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }
    //upis nizova
    cout<<"\n";
    for(i=0;i<m;i++)
    {
        cout<<n<<" elemenata "<<i+1<<". niza: ";
        for(j=0;j<n;j++)
            cin>>niz[j];

        f.write((char*) niz,n*sizeof(float));
    }
    //zatvaranje datoteke
    f.close();
}

void razvrst(char *ime, int n, int m)
{
    int i,j,brojp, brojn;
    float niz[MAX];

    //otvaranje binarne datoteke za citanje
    f.open(ime,ios::in|ios::binary);
    if(!f)
    {
        cout<<"\ngreska kod otvaranja datoteke\n";
        exit(1);
    }

    //otvaranje tri binarne datoteke
    f1.open("poz.bin",ios::out|ios::binary);
    if(!f1)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }
    f2.open("neg.bin",ios::out|ios::binary);
    if(!f2)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }
    f3.open("nula.bin",ios::out|ios::binary);
    if(!f3)
    {
        cout<<"\n Nije uspelo otvaranje datoteke\n";
        exit(1);
    }
}

```

```

for(i=0;i<m;i++)
{
    brojn=brojp=0;        //pretpostavka: isti je broj neg. i poz. elemenata

    f.read((char*)niz,n*sizeof(float));

    for(j=0;j<n;j++)
        if(niz[j]<0)
            brojn++;
        else if(niz[j]>0)
            brojp++;

    if(brojp>n-brojn)
    {
        f1.write((char*) niz,n*sizeof(float));
        bpoz++;        //jos jedan niz sa najvise pozitivnih elemenata
    }
    else if(brojn>n-brojn)
    {
        f2.write((char*) niz,n*sizeof(float));
        bneg++;        //jos jedan niz sa najvise negativnih elemenata
    }
    else
    {
        f3.write((char*) niz,n*sizeof(float));
        bnula++;        //jos jedan niz sa najvise nula elemenata
    }
}

//zatvaranje datoteka
f.close();f1.close();f2.close();f3.close();
}

void meni(int n)
{
    int opcija;

    cout<<"\nlzaberite opciju:\n1. pozitivni\n2. negativni\n3.nula\n";
    cin>>opcija;

    cout<<"\nSadrzaj datoteke: "<<endl;
    switch(opcija)
    {
        case 1:
            prikaz("poz.bin",bpoz,n);
            break;
        case 2:
            prikaz("neg.bin",bneg,n);
            break;
        case 3:
            prikaz("nula.bin",bnula,n);
            break;
        default:
            cout<<"Nije zadata ni jedna od mogucih opcija!\n";
            break;
    }
}
}

```

```

void prikaz(char *ime, int broj,int n)
{
    fstream f;
    int i,j;
    float niz[MAX];

    //otvaranje binarne datoteke za citanje
    f.open(ime,ios::in|ios::binary);
    if(!f)
    {
        cout<<"\ngreska kod otvaranja datoteke\n";
        exit(1);
    }

    f.seekg(0);
    for(i=0;i<broj;i++)
    {
        f.read((char*)niz, n*sizeof(float));

        cout<<endl;
        for(j=0;j<n;j++)
            cout<<niz[j]<<"\t";
    }
    cout<<endl;

    //zatvaranje datoteke
    f.close();
}

```

Задатак 5_9.

Написати програм на језику C++ који:

- динамички додељује меморију за списак који треба да чува податке за највише 500 станова: *ulica*, *broj*, *broj_stana*, *broj_soba* и *broj_kvadrata*,
- чита са тастатуре податке за сваки стан и формира од ових података запис у бинарној датотеци *stanari.bin*,
- приказује на екрану списак свих станова на адреси (*ulica*, *broj*) задатој преко тастатуре.

Решење 5_9.

```

#include<fstream.h>
#include<stdlib.h>
#include <string.h>
#define BROJ 150          //dozvoljen broj struktura u spisku
#define MAX 30            //broj znakova u nazivima
struct stan{
    char ulica[MAX+1];
    int broj, broj_stana,broj_soba;
    float kvadrata;
};

```



```

fstream fptr;                //kreiranje objekta za rad sa datotekom

void upis(char *ime, struct stan niz[], int n);
void pretraga(char *ime, struct stan niz[], int n);

main()
{
    int n;
    struct stan *niz;

    do
    {
        cout<<"\nBroj stanova (najvice "<<MAX<<") stanova: ";
        cin>>n;
    }while(n<0 || n>MAX);

    //dodela memorije
    niz=new struct stan [n];
    if(!niz)
    {
        cout<<"\nNije uspelo otvaranje memorije\n";
        exit(1);
    }

    upis("stanari.bin", niz, n);
    pretraga("stanari.bin", niz, n);

    //oslobadjanje memorije
    delete [] niz;

    return 0;
}

void upis(char *ime, struct stan niz[], int n)
{
    cout<<"\nUnos podataka za stanove\n";
    for(int i=0;i<n;i++)
    {
        cin.ignore();                //citanje zaostalih karaktera

        cout<<"\nUlica i broj: ";
        cin>>niz[i].ulica>>niz[i].broj;

        cout<<"\nBroj stana, broj soba i broj kvadrata: ";
        cin>>niz[i].broj_stana>>niz[i].broj_soba>>niz[i].kvadrata;
    }

    //otvaranje datoteke za upis
    fptr.open(ime, ios::out|ios::binary);
    if(!fptr)
    {
        cout<<"\nNije uspelo otvaranje datoteke\n";
        exit(1);
    }
}

```

```

//upis niza struktura u datoteku
fptr.write((char*)niz,n*sizeof(struct stan));

//zatvaranje datoteke
fptr.close();
}

void pretraga(char *ime, struct stan niz[], int n)
{
    int broj;
    char ulica[MAX+1];
    struct stan temp;

    while(1)
    {
        //otvaranje datoteke za citanje
        fptr.open(ime,ios::in|ios::nocreate);
        if(!fptr)
        {
            cout<<"\nNije uspelo otvaranje datoteke\n";
            exit(1);
        }

        //zadavanje podataka za pretragu
        cout<<"\nPretraga datoteke\n";
        cout<<"\nUlica i broj: ";
        cin>>ulica>>broj;

        int nadjen=0;           //na pocetku pretrage status nadjen je 0

        //pretraga niza struktura iz datoteke
        for(int i=0; i<n;i++)
        {
            fptr.read((char*)&temp,sizeof(struct stan));
            if(strcmp(temp.ulica,ulica)==0 && temp.broj==broj)
            {
                cout<<temp.ulica<<" "<<temp.broj<<"/"<<temp.broj_stana;
                cout<<" , soba: "<<temp.broj_soba;
                cout<<" , kvadrata: "<<temp.kvadrata<<endl;
                nadjen=1;
            }
        }

        //ako se zada nepostojeca adresa, onda je kraj petlje
        if(!nadjen)
        {
            cout<<"\nNema stanova na trazenoj adresi\n";
            break;
        }
        //zatvaranje datoteke
        fptr.close();
    }
}

```

Задатак 5_10.

Написати програм на језику C++ који:

- а) чита са тастатуре податке за n ($n \leq 500$) књига (*naziv_knjige*, *datum_izdavanja* у облику gggg.mm.dd. и *ime_citaoca*) и од њих и аутоматски генерисаног податка *redni_broj* формира записе у бинарној датотеци *biblioteka.bin*,
- б) од свих записа редом прочитаних из датотеке *biblioteka.bin* издваја оне код којих је *datum_izdavanja* старији од датума задатог преко тастатуре (у облику gggg.mm.dd.) и формира од њих записе у текстуалној датотеци *izbor.txt*,
- в) приказује на екрану садржај текстуалне датотеке *izbor.txt*.

Решење 5_10.

```
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#define MAX2 500           //dozvoljen broj knjiga
#define MAX1 20            //broj znakova u nazivima

struct knjiga{             //definicija tipa strukture
    int redni_broj;
    char naziv_knjige[MAX1+1];
    char datum[MAX1+1];
    char ime_citaoca[MAX1+1];
};

int broj_zapisa_b, broj_zapisa_t; //brojevi zapisa u binarnoj i tekst datoteci
fstream fptr1, fptr2;           //kreiranje objekata za rad sa datotekama

void formira(void);
void izbor(void);
void prikaz(void);

main()
{
    //otvaranje svih potrebnih datoteka
    fptr1.open("biblioteka.bin", ios::out | ios::in | ios::binary);
    if(!fptr1)
    {
        cerr << "Nije uspjelo otvaranje datoteke";
        exit(1);
    }
    fptr2.open("izbor.txt", ios::out | ios::in);
    if(!fptr2)
    {
        cerr << "Nije uspjelo otvaranje datoteke";
        exit(1);
    }

    formira();
    izbor();
    prikaz();
}
```

```

        //zatvaranje datoteka
        fptr1.close();
        fptr2.close();

        return 0;
    }
    void formira(void)
    {
        int i;
        struct knjiga zapis;

        do
        {
            cout << "\nUneti broj knjiga: ";
            cin >> broj_zapisa_b;
        }while(broj_zapisa_b<1 || broj_zapisa_b>MAX2);

        //citanje podataka sa tastature i upis u binarnu datoteku
        cout << "\nUneti za svaku knjigu, u posebnoj liniji";
        cout << "\nnaziv knjige, datum izdavanja (gggg.mm.dd.) i ime citaoca:\n";
        for(i=0; i<broj_zapisa_b; i++)
        {
            zapis.redni_broj = i+1;

            cin >> zapis.naziv_knjige >> zapis.datum >> zapis.ime_citaoca;
            fptr1.write((char *)&zapis,sizeof(knjiga));
        }
    }
    void izbor(void)
    {
        int i;
        char datum_min[MAX1+1];
        struct knjiga zapis;

        cout << "\nUneti datum (gggg.mm.dd.): ";
        cin >> datum_min;

        fptr1.seekg(0);           //povratak na pocetak binarne datoteke
        broj_zapisa_t=0;

        for(i=0; i<broj_zapisa_b; i++)
        {
            zapis.redni_broj=i+1;

            //citanje iz binarne datoteke
            fptr1.read((char *)&zapis,sizeof(knjiga));

            //upis pojedinih zapisa u tekst datoteku
            if( strcmp(zapis.datum,datum_min)<0)
            {
                broj_zapisa_t++;
                fptr2 << zapis.redni_broj << ' ' << zapis.naziv_knjige << ' '
                    << zapis.datum << ' ' << zapis.ime_citaoca <<'\n';
            }
        }
    }
}

```

```

void prikaz(void)
{
    int i;
    struct knjiga zapis;

    fptr2.seekg(0);

    //prikaz na ekranu sadrzaja tekst datoteke
    cout << "\nSpisak knjiga izdatih pre zadatog datuma:\n";
    for(i=0; i<broj_zapisa_t; i++)
    {
        fptr2 >> zapis.redni_broj
                >> zapis.naziv_knjige
                >> zapis.datum
                >> zapis.ime_citaoca;

        cout << zapis.redni_broj << ' '
                << zapis.naziv_knjige << ' '
                << zapis.datum << ' '
                << zapis.ime_citaoca << "\n";
    }
}

```


Литература

- [1] Ласло Краус, Програмски језик C са решеним задацима, пето издање, Академска мисао, Београд 2006.
- [2] Ласло Краус, Програмски језик C++ са решеним задацима, пето издање, Академска мисао, Београд 2004.
- [3] Слободан Обрадовић, Вештина доброг програмирања, Виша електротехничка школа, Београд 2004.
- [4] Милан Чабаркапа, C основи програмирања, Круг, Београд 1996.
- [5] *Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language - ANSI C, Prentice Hall Software Series, 1988.*
- [6] *Augie Hansen, Програмирање на језику C – потпуни водич за програмски језик C, Микро књига, Београд 1991.*
- [7] *Peter Aitken, Bradley Jones, Teach Yourself C in 21 Days, CD edition, 1998.*
- [8] *Chris H. Pappas, William H. Murray, C/C++ Водич за програмере, Микро књига, Београд 1996.*