Основи програмирања

Предавање 2.

Основни типови података у језику "C"

Висока школа електротехнике и рачунарства у Београду

Основи типови података

- У језику C:

  - дефинисани су само нумерички типови података

  - два основна начина за чување ових података

    ПРОМЕНЉИВЕ и КОНСТАНТЕ

Променљиве

- Локације у меморији:
  - које се резервишу из програма
  - имају ИМЕ
  - имају ТИП

- Име променљиве је идентификатор језика C:
  - слова, цифре и знак доња црта (_)
  - први знак – слово или доња црта
  - службене (резервисане) речи - не
  - прави се разлика - велика / мала слова

Службене речи програмског језика C

Службене (резервисане) речи језика "C"			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Примери са разликама велика / мала слова у додељеним именима променљивама

ime
Ime
IME

три различита имена у *C* програму

x_koordinata
x_Koordinata
X_koordinata
X_Koordinata
X_KOORDINATA

пет различитих имена у *C* програму

Примери исправно и неисправно додељених имена променљивама

исправна имена

i

ime

ime_2

ime_novo (under_line)

NekoIme (PascalCase)

nekoNovoIme (camelCase)

начини доделе кад име садржи више речи

неисправна имена

#neko_ime (знак #)

ime:1 (знак :)

2ime (на 1. месту цифра)

ime 3 (празан знак)

break (резервисана *C* реч)

while (резервисана *C* реч)

Променљиве

- Типови променљивих:
  - ЦЕЛОБРОЈНИ
  - РЕАЛНИ
- Подтипови:
  - целобројни – за различите опсеге
  - реални – за различите прецизности

Основни означени целобројни типови

Тип	Ознака	Величина (у бајтовима)	Опсег
Character	char	1	-128 до 127
Short	short	2	-32768 до 32767(1)
Integer	int	2 / 4 /	(1) / (2) /
Long	long	4	-2 147 483 648 до 2 147 438 647 (2)

Релације између величина појединих типова

sizeof(char) < sizeof(short) ≤ sizeof(int) ≤ sizeof(long)

Основни неозначени целобројни типови

Тип	Ознака	Величина (у бајтовима)	Опсег
Unsigned Character	unsigned char	1	0 do 255
Unsigned Short	unsigned short	2	0 do 65535 (3)
Unsigned Integer	unsigned	2 / 4	(3) / (4)
Unsigned Long	unsigned long	4	0 до 4,294,967,295 (4)

sizeof(unsigned char) < sizeof(unsigned short) ≤ sizeof(unsigned) ≤ sizeof(unsigned long)

Основни реални типови

Тип	Ознака	Величина (у бајтовима)	Опсег
Single-precision Floating_point	float	4	1.2E-38 до 3.4E38 (прецизност=7 цифара)
Double-precision Floating-point	double	8	2.2E-308 до 1.8E308 (прецизност=16 цифара)

Релације између величина реалних типова

sizeof(float) < sizeof(double)

Оператор sizeof()

- Примена оператора sizeof(tip) над основним типом даје величину у бајтовима тог типа, пример:

sizeof(char) → 1

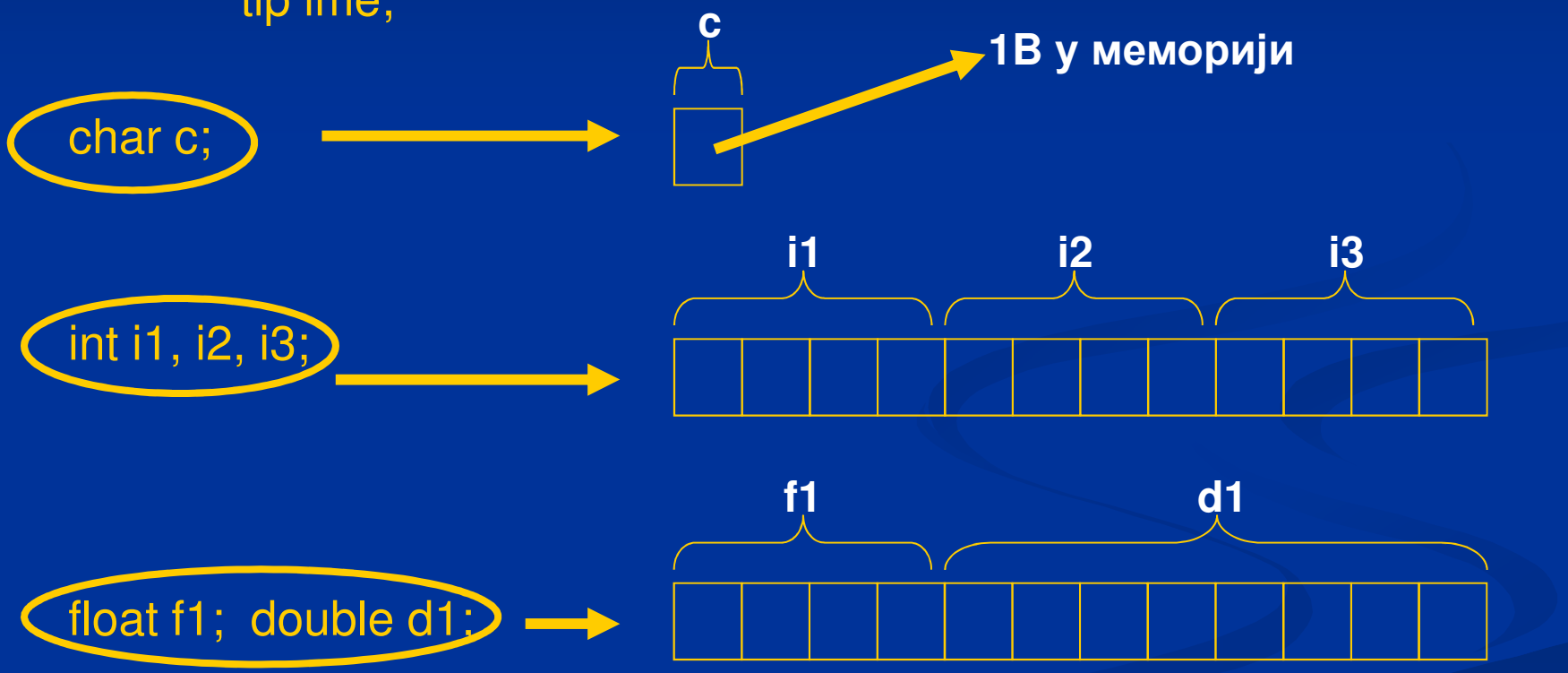
sizeof(int) → 2/4

sizeof(double) → 8

Декларација променљивих

- Декларација променљивих:

  tip ime;



Иницијализација променљивих

- Унутар декларације

  - ако је иницијална вредност променљиве потребна од почетка програма / функције

```
int i1=10, i2=10, i3=100;
```

~~int i1=i2=10, i3=100;~~

- Ван декларације

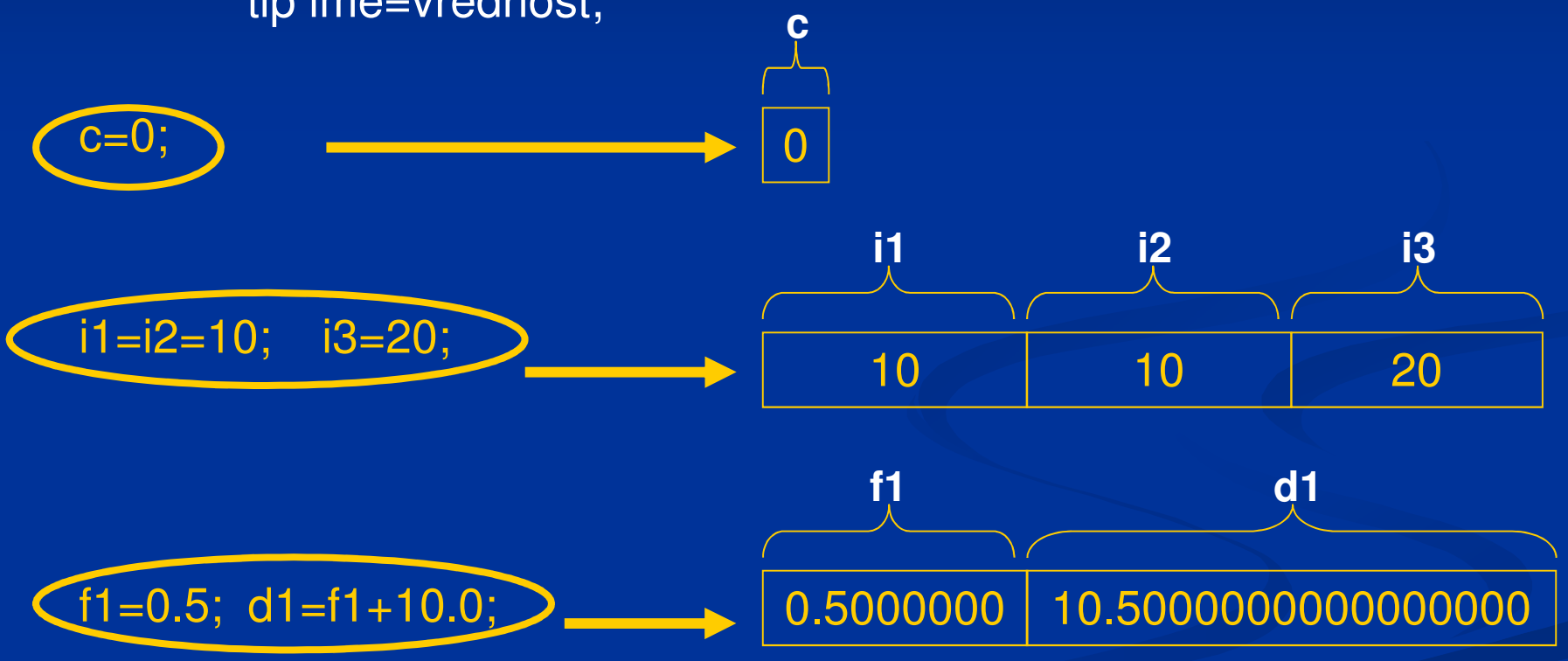
  - ако иницијална вредност променљиве није неопходна од почетка програма / функције:

```
int i1, i2, i3;
…..
i1=i2=10;          i3=100;
```

Иницијализација променљивих

- Иницијализација променљивих:

  tip ime=vrednost;



Константе

Два начина увођења константи у програм:

1. ДЕКЛАРАЦИЈА КОНСТАНТИ
   - при извршавању програма заузима локацију у меморији
   - има тип и име

2. ДЕФИНИЦИЈА КОНСТАНТИ
   - нема локацију у меморији
   - има само име, нема дефинисани тип

Декларисане константе

- Локације у меморији:
  - које се резервишу из програма
  - имају ИМЕ
  - имају ТИП
  - имају ВРЕДНОСТ

- Име константе је идентификатор језика C:
  - слова, цифре и знак доња црта (_)
  - први знак – слово или доња црта
  - службене речи - не
  - прави се разлика - велика / мала слова

исто као за променљиве

Декларисане константе

- Тип константе:
  - ЦЕЛОБРОЈНИ
  - РЕАЛНИ
  - ЗНАКОВНИ
- Подтипови:
  - целобројни – за различите опсеге
  - реални – за различите прецизности

Декларисане константе

- Декларација константи:

```
const tip ime=vrednost;
```

```
const int ceo_broj=100;
```

ceo_broj

100

```
const float real_broj=0.5;
```

real_broj

0.5000000

```
const char slovo='A';
```

65

slovo

(ASCII број)

Коришћење променљивих и константи

- Декларација променљивих – дозвољава измену вредности

```
int i1=0, i2=10, i3=100;
i3=i1+i2;
```

- Декларација константи – не дозвољава измену вредности

```
const int i1=0, i2=10, i3=100;
i3=i1+i2;
```

Дефинисане константе
- СИМБОЛИЧКЕ КОНСТАНТЕ -

- Нема локације у меморији
  - постоји СИМБОЛ
  - постоји ВРЕДНОСТ – целобројна, реална или знаковна

- Име константе је идентификатор језика C:
  - слова, цифре и знак доња црта (_)
  - први знак – слово или доња црта
  - службене речи - не
  - прави се разлика - велика / мала слова

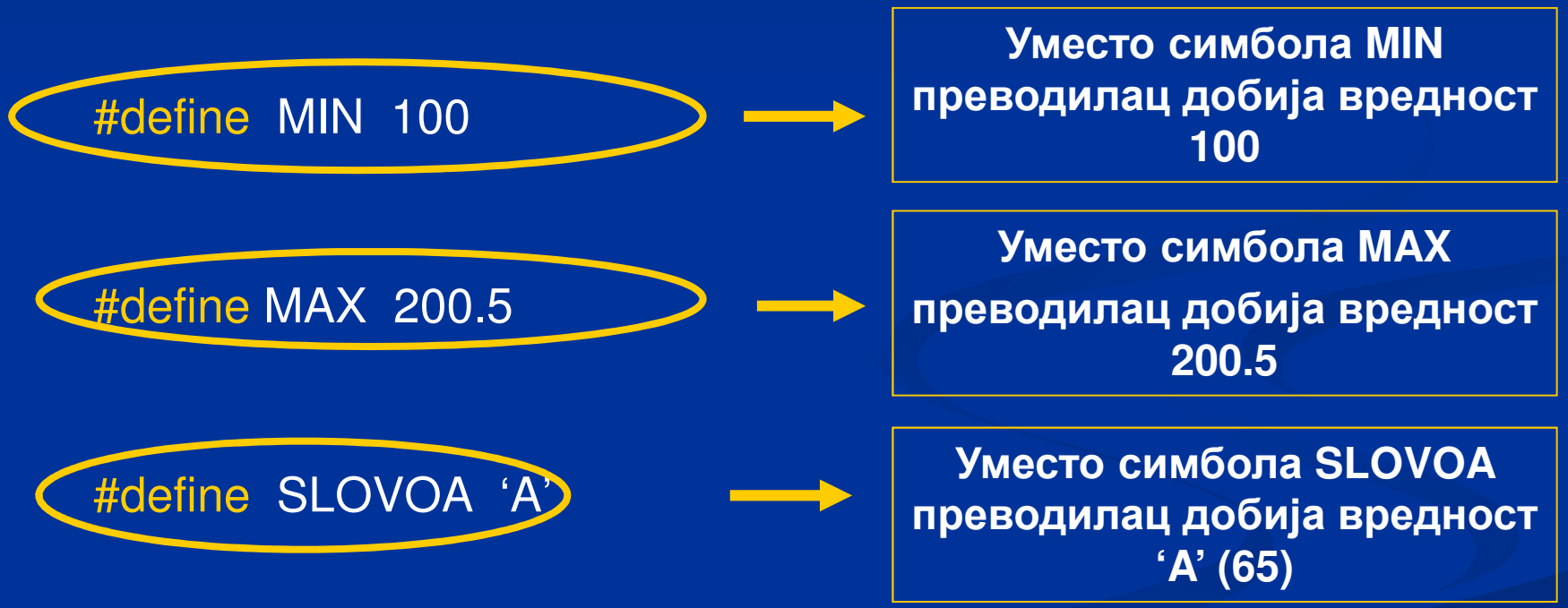
исто као за променљиве

УОБИЧАЈЕНО – ВЕЛИКА СЛОВА

Симболичке константе

- Дефиниција константи:

  # define SIMBOL vrednost



Симболичке знаковне константе

```
#define  U1   '\t'
#define  U2  '\n'

#define  Z1   ' '
#define  Z2   '*'
#define  Z3   '.'

#define  C1  '0'
#define  C2  '9'

#define  S1  'A'
#define  S2  'Z'

#define  S3  'a'
#define  S4  'z'
```

```
#define U1 9
#define U2 10

#define Z1 32
#define Z2 42
#define Z3 46

#define  C1 48
#define  C2  57

#define  S1  65
#define  S2  90

#define  S3  97
#define  S4  122
```

Групе знакова

I

II

III

IV

V

Симболичке знаковне константе

```
#define  TAB   '\t'          /*jedna znakovna konstanta*/
#define  EOL   '\n'          /*jos jedna znakovna konstanta*/

/*Konstantni niz znakova izmedju navodnika*/
#define PORUKA “Greska u otvaranju Datoteke!“
```

```
#define  TAB   '\t' ;
#define  EOL   '\n' ;
```


Грешка, претпроцесор симбол TAB замењује низом знакова '\t';

(знак ; је вишак)

Грешка, претпроцесор симбол EOL замењује низом знакова '\n';

(знак ; је вишак)

Употреба симболичких константи

```
#define  TRUE   1
#define  FALSE  0
```

→ **За тип bool (true / false) који није дефинисан у језику C**

```
#define  MIN  -100
#define  MAX  100
```

→ **За опсеге бројева који су често изменљиви**

```
#define  PI  3.141593
```

→ **И за константе које имају предефинисане вредности**

Избор начина увођења константи у програм

```
#define  PI  3.141593
```

→ **Упутство претпроцесору**
Нема заузећа у меморији
Нема провере типова

```
const float  pi = 3.141593;
```

→ **Упутство за преводиоца**
Заузеће у меморији
Проверава се тип

Дефиниција набројаних константи

- Дефиниција набројаних константи:

  enum ime {IME1=vrednost1, IME2=vrednost2, … , IMEn=vrednostn};

- Замена за више директива:

  #define IME1 vrednost1
  #define IME2 vrednost2
  …
  #define IMEn vrednostn

Само ако су константе целобројне

Дефиниција набројаних константи

- Примери:

enum stanje {OFF, ON}; → **Подразумевано OFF:0,ON:1**

enum dani {PON=1, UTO, SRE, CET, PET, SUB, NED};

↓

Подразумевано UTO:2, SRE:3, CET:4, PET:5, SUB:6, NED:7

enum opseg {MIN=-10, NUL=0, MAX=10};

↓

Све константе су дефинисане, нема подразумеваних

Декларација променљивих набројаног типа

- Начин 1.

```
enum dani { PON=1, UTO, SRE, CET, PET, SUB, NED } danas;
danas=PON;
```

Променљива danas типа enum dani

Променљива danas добија вредност 1

- Начин 2.

```
enum dani { PON=1, UTO, SRE, CET, PET, SUB, NED };
enum dani sutra=UTO;
```

Променљива sutra типа enum dani добија вредност 2

Дефинисање синонима за било који тип језика C

typedef tip sinonim;

```
typedef char znak;
typedef int ceo_broj;
typedef float real_broj;
```

Дефиниција синонима

```
znak c = 's';
ceo_broj x = 100;
real_broj y = 0.25;
```

Коришћење синонима

Форматиран улаз / излаз података

- Форматиран излаз

  printf(“ formati ”, imena promenljivih);

- Форматиран улаз

  scanf(“ formati ”, adrese promenljivih);

- излазни и улазни формати:

“ ...format ...”  **“ … %slovo …”**

Форматирани ИЗЛАЗ података

- Излазни формати за основне типове (у функцији printf()):

  %c (char)

  %d (int – децимални облик)

  %x (X) (int / unsigned int – хексадецимални облик)

  %o (int / unsigned int – октални облик)

  %u (unsigned int – децимални облик)

  %hd, %hx, %ho (short int – у сва три облика)

  %ld, %lx, %lo (long int – у сва три облика)

  %f (float / double – облик са децималном тачком)

  %e (float / double – облик са експонентом)

  %g (float / double – најједноставнији облик)

Форматирани УЛАЗ података

- Улазни формати за основне типове (у функцији scanf()):
  %c (char)
  %d (int – децимални облик)
  %x (X) (int / unsigned int – хексадецимални облик)
  %o (int / unsigned int – октални облик)
  %u (unsigned int – децимални облик)

  %hd, %hx, %ho (short int – у сва три облика)
  %ld, %lx, %lo (long int – у сва три облика)

  %i (децимални, хексадецимални или октални облик
  облик је неопходно назначити: /, 0x(0X) или 0)

  %f , %e, %g (float – одговарајући облик)
  %lf, %le, %lg (double – одговарајући облик)

Форматиран ИЗЛАЗ

```
#include <stdio.h>
main()
{  int x=10;
   float y=20.5;

   printf( “Formatirani izlaz:\n");
   printf( “%d\n", x );
   printf( "%d%f \n“, x, y );
   printf( "%d %f \n“, x, y );
   printf( "%d  %x %o \n“, x, x, x );
   printf( “Broj x ima vrednost: %d\n“, x);
   printf( “Broj y ima vrednost: %f\n“, y);
}
```

излазни уређај

```
C:\>Formatirani izlaz:
C:\>10
C:\>1020.500000
C:\>10 20.500000
C:\>10 a 12
```

```
C:\>Broj x ima vrednost 10
C:\>Broj y ima vrednost 20.500000
```

Формати ИЗЛАЗА различити од подразумеваних

Улазни формати за основне типове (у функцији printf()):

% [- + 0 # n . m] slovo

Форматиран ИЗЛАЗ
са форматима различитим од подразумеваних

```
#include <stdio.h>

main()
{  int x=5;
   float y=20.5;

   printf( “%10d \n", x );
   printf( "%010d \n“, x );
   printf( "%-10d \n“, x );
   printf( "%.4f \n“, y );
   printf( “%10.4f \n“, y);
   printf( “%e \n“, y);
}
```

излазни уређај

```
C:\>          5_
C:\>0000000005_
C:\>5         _
C:\>20.5000_
C:\>    20.5000_
C:\>2.050000e+001_
```

Форматиран УЛАЗ

```
#include <stdio.h>

main()
{
   int x, x1, x2, x3;
   float y;
   double z;

   scanf( " %d ", &x );
   scanf( " %d%f%lf ", &x, &y, &z );
   scanf( "%d:%f:%lf", &x, &y, &z );

   scanf( "%d%x%o ", &x1, &x2, &x3);
   scanf( "%i%i%i", &x1, &x2, &x3);
}
```

ул. уређај и меморија

C:\>1					
x	x1	x2	x3	y	z
1					

C:\>2 0.2 0.5					
x	x1	x2	x3	y	z
2				0.2	0.5

C:\>3:1.5:2.5					
x	x1	x2	x3	y	z
3				1.5	2.5

C:\>10 a 12					
C:\>10 0xa 012					
x	x1	x2	x3	y	z
3	10	10	10	1.5	2.5

Формати УЛАЗА различити од подразумеваних

Улазни формати за основне типове (у функцији scanf()):

% [n] slovo

Максимални број цифара који се прихвата од броја

Форматиран УЛАЗ
са форматима различитим од подразумеваних

```
#include <stdio.h>

main()
{
   int x1, x2, x3;
   float y1,y2;

   scanf( “%2d%2d%d”, &x1, &x2, &x3);
   scanf( “%2d%2d%d”, &x1,&x2,&x3 );

   scanf( “%2f “, &y1 );
   scanf( “%f “, &y2 );
}
```

ул. уређај и меморија

C:\>10 20 30				
x1	x2	x3	y1	y2
10	20	30		

C:\>12560				
x1	x2	x3	y1	y2
12	56	0		

C:\>2530.25				
x1	x2	x3	y1	y2
12	56	0	25.00	30.25

Припрема лабораторијске вежбе код куће

- Лекције из уџбеника
  (поглавље 2. Подаци)
- Анализа примера и задатака у вежби 2. у збирци
  (Основни типови података у језику C)
- Израда задатака
  за припрему лабораторијске вежбе 2. у збирци

Основи програмирања

Предавање 2.

Основни типови података у језику "C"

Висока школа електротехнике и рачунарства у Београду