

· Република Србија ·  
· Академија техничко - уметничких струковних студија Београд ·  
· Одсек Висока школа електротехнике и рачунарства ·  
· 2026 ·



## ДЕСЕТО ДРЖАВНО ТАКМИЧЕЊЕ ИЗ ПРОГРАМИРАЊА

### ТЕКСТОВИ ЗАДАТАКА

НА СРПСКОМ ЈЕЗИКУ

Три задатка · 180 минута · 150 бодова укупно

### ПРАВИЛА ТАКМИЧЕЊА

- Сваки задатак носи максимално **50 бодова**.
- На почетку сваког задатка уносе се потребне улазне вредности.
- Сваки задатак има **5 излаза** који се рачунају на основу улазних вредности унетих на почетку.
- Сваки излаз у задатку носи **10 бодова** ако је тачан, **0** ако је нетачан.
- Такмичари су обавезни да **ЈАСНО** назначе сваки излаз ознаком подзадатка (нпр. **1.1:**, **1.2:** итд.). Ако излаз није јасно назначен такмичар добија 0 поена за тај излаз.
- Ако задатак престане да ради због грешке у било ком тренутку, такмичару се бодују излази који су до тог тренутка видљиви на екрану.
- У случају истог броја бодова, бољи пласман остварује такмичар који је раније предао задатак.
- Такмичење траје **180 минута**.

## ЗАДАТАК 1

Унети два петоцифрена броја **a** и **b** (одвојена размаком, у једном реду). Претпоставка јесте да су бројеви исправно унети, односно да није потребно проверавати да ли су унети бројеви заиста петоцифрени. Сваки подзадатак има тачно **један ред излаза** – јасно га означити пре исписивања (нпр. **1.1:...**).

**A****Суме, производи и разлика**

Исписати у једном реду пет бројева: суму цифара броја **a**, суму цифара броја **b**, производ цифара броја **a**, производ цифара броја **b**, и апсолутну вредност разлике суме цифара броја **a** и суме цифара броја **b** (тим редоследом, одвојене размацима).

**Б****Формирање новог броја c**

Формирати нови петоцифрени број **c** тако што се свака цифра на позицији  $i$  ( $i = 0$  за цифру јединица,  $i = 1$  за цифру десетица,  $i = 2$  за цифру стотица,  $i = 3$  за цифру хиљада,  $i = 4$  за цифру десетхиљада) рачуна по формули:

$$c[i] = (a[i] + b[i] + i) \bmod 10$$

Исписати само број **c**.

**В****Низ пребројавања цифара**

Формирати низ **d** од 10 елемената тако да **d[i]** представља укупан број појављивања цифре  $i$  у бројевима **a**, **b** и **c** заједно. Исписати елементе низа **d** у једном реду, одвојене размацима (тачно 10 бројева, од **d[0]** до **d[9]**).

**Г****Матрица mat – централни елемент**

Формирати матрицу **mat** димензија  $10 \times 10$  по следећим правилима:

- Први ред (ред 0) је низ **d**.
- Сваки наредни ред  $i$  (за  $1 \leq i \leq 9$ ) се добија од реда  $i-1$  тако што се сви елементи претходног реда ротирају циклично **улево** за једно место, а затим се сваком елементу дода вредност  $i$  уз задржавање само цифре јединица резултата (тј.  $n \bmod 10$ ).

Исписати само вредност централног елемента: **mat[5][5]**.

**Д****Ротација матрице – елемент rot[2][8]**

Формирати матрицу **rot** тако што се матрица **mat** ротира за  $90^\circ$  у смеру казаљке на сату. Исписати само вредност елемента **rot[2][8]**.

## ЗАДАТАК 2

Унети стринг **s1** дужине **3·N** ( $3 \leq N \leq 12$ ) карактера, који се састоји од малих и великих слова енглеског алфавета и цифара. Претпоставка јесте да је стринг исправно унет, односно да није потребно то проверавати. **N** није посебно унет – треба га израчунати из дужине стринга. Индекс 0 одговара прваком карактеру стринга (нпр. **s1[0]** је први карактер). Сваки подзадатак има тачно **један ред излаза** – јасно га означити пре исписивања (нпр. **2.1:...**).

**A** Основна статистика и уређеност

У једном реду исписати четири вредности (одвојене размацима): број великих слова, број малих слова, број цифара, и реч **DA** ако је стринг **уређен**, у супротном **NE**. Уређеним се сматра стринг у којем не постоје **два узастопна карактера истог типа** (типови: мало слово, велико слово, цифра).

**Б** Три секвенце – два карактера

Од стринга **s1** формирати 3 секвенце, сваку дужине **N**, тако да:

- **sek1** садржи карактере са позиција 0, 3, 6, 9, ... (остатак 0 при дељењу са 3),
- **sek2** садржи карактере са позиција 1, 4, 7, 10, ... (остатак 1 при дељењу са 3),
- **sek3** садржи карактере са позиција 2, 5, 8, 11, ... (остатак 2 при дељењу са 3).

Исписати у једном реду **тачно два карактера** одвојена размаком: последњи карактер прве секвенце и први карактер треће секвенце. Формат: **sek1[N-1] sek3[0]**.

**В** Трансформација карактера – два карактера

У свакој од три секвенце извршити следећу трансформацију карактера:

- **Мала слова** померити циклично за **3 места унапред** у енглеском алфавету ('a' → 'd', 'b' → 'e', ..., 'w' → 'z', 'x' → 'a', 'y' → 'b', 'z' → 'c').
- **Велика слова** заменити одговарајућим малим словима ('A' → 'a', 'B' → 'b', ..., 'Z' → 'z').
- **Цифру** заменити следећом цифром циклично ('0' → '1', '1' → '2', ..., '8' → '9', '9' → '0').

Трансформација се примењује независно на сваки карактер секвенце. Трансформисана секвенца добијена од **sek1** означава се са **t1**, од **sek2** са **t2**, а од **sek3** са **t3**. Исписати у једном реду **тачно два карактера** одвојена размаком: **t1[N-1] t3[0]**.

**Г** Спајање и фреквенце карактера

Спојити трансформисане секвенце (редом: прва, друга, трећа) у један стринг **s4**. У једном реду исписати **четири вредности** одвојене размацима: карактер који се у **s4** појављује **највећи** број пута, тај број појављивања, карактер који се појављује **најмањи** број пута, и тај број појављивања. Разматрају се само карактери који се у **s4** појављују барем једанпут. Уколико постоји више таквих карактера, узети онај са **најмањом ASCII вредношћу**. Формат: **max\_kar max\_broj min\_kar min\_broj**.

## Д

Квадратна матрица и ротација – елементи `mat[0][K-1]` и `rot[K-1][0]`

Одредити  $K = \text{ceil}(\text{sqrt}(\text{len}(s4)))$  – димензију најмање квадратне матрице `mat` у коју се `s4` може уписати ред по ред (где `len(s4)` означава дужину стринга `s4`, а `ceil` и `sqrt` су функције заокружавања навише и квадратног корена). Празна места попунити карактером '@'. Затим формирати матрицу `rot` тако што се матрица `mat` ротира за  $90^\circ$  улево (у смеру супротном казаљки на сату), по формули `rot[i][j] = mat[j][K-1-i]`. У једном реду исписати **три вредности** одвојене размацама: вредност `K`, затим елемент `mat[0][K-1]`, па елемент `rot[K-1][0]`. Формат: `K mat[0][K-1] rot[K-1][0]`.

## ЗАДАТАК 3

Пингвин **Пинго** се нашао у леденој области коју држи стари морски лав **Лео**. Област је представљена матрицом  $10 \times 10$  поља. Координатни почетак је у горњем левом углу матрице (поље (0,0)) – **редови и колоне су индексирани од 0 до 9**.

Унети четири једноцифрена броја **r1 c1 r2 c2** (одвојена размацама, у једном реду). Тиме су задате две позиције:

- **(r1, c1)** – почетна позиција **Пинга** (пингвина),
- **(r2, c2)** – почетна позиција **Леа** (морског лава).

Сваки подзадатак има тачно **један ред излаза**. Само подзадатак **3.5** додатно црта матрицу.

## Шта је Менхетн растојање?

За две тачке  $(x_1, y_1)$  и  $(x_2, y_2)$  у матричној мрежи, **Менхетн растојање** је збир апсолутних разлика по координатама:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

То је **број корака** потребан да се од прве тачке стигне до друге ако се крећемо само хоризонтално или вертикално (никада дијагонално) – замислите кретање улицама у граду са правоугаоним блоковима, отуда и назив (по острву Менхетн у граду Њујорку).

**Пример:** између тачака (3, 3) и (7, 7) Менхетн растојање је  $|3 - 7| + |3 - 7| = 4 + 4 = 8$  корака (а не  $\sqrt{32} \approx 5,66$  као што би било код правог еуклидског растојања).

## Правила игре (поједностављена):

- Пинго иде у правцу **најближе ивице матрице** (по Менхетн растојању од сваке стране). Ако је подједнако удаљен од више ивица, бира смер по приоритету: **ГОРЕ** (ка реду 0), **ДЕСНО** (ка колони 9), **ДОЛЕ** (ка реду 9), **ЛЕВО** (ка колони 0). Креће се по правој линији у том смеру, једно поље по потезу.
- Пинго се сматра да је **побегао** ако стигне до било које ивице матрице.
- Лео се не помера у овом једноставнијем моделу. Поређењем растојања се одређује исход:
  - ако је **растојање Пинга до најближе ивице**  $\leq$  **Менхетн растојање Пинга и Леа**, Пинго побегне (стиже до ивице пре него што Лео може да га ухвати),
  - у супротном, Лео хвата Пинга на његовој почетној позицији **(r1, c1)**.

**A** Менхетн растојање

Исписати Менхетн растојање између почетних позиција Пинга и Леа.

**Б** Удаљеност Пинга од најближе ивице

Исписати најмању Менхетн удаљеност Пинга од било које ивице матрице ( $\min(r1, 9-r1, c1, 9-c1)$ )

**В** Смер кретања Пинга

Исписати реч која означава смер у ком ће се Пинго кретати, према приоритету: **GORE**, **DESN0**, **DOLE** или **LEVO**. Ако је Пинго већ на ивици (растојање = 0), исписати **IVICA**.

**Г****Исход и крајња позиција**

У једном реду исписати исход и финалну позицију Пинга. Формат:

- ако је побегао: **PINGVIN POBEGAO NA (X,Y)**,
- ако је ухваћен: **LAV UHVATIO NA (X,Y)**.

**Д****Матрица одговора са лабелама**

Исписати матрицу **10×10** са **ознакама редова и колона** (бројеви 0–9).

Користити симболе:

- **'P'** – почетна позиција Пинга (поље **(r1, c1)**),
- **'L'** – позиција Леа (поље **(r2, c2)**),
- **'p'** – путања кроз коју је Пинго прошао (без почетне и крајње позиције),
- **'F'** – крајње поље ако је Пинго побегао (на ивици),
- **'X'** – поље на ком је Лео ухватио Пинга (поклапа се са **'P'** – у том случају пише само **'X'**),
- **'-'** – сва остала поља.

Формат исписа: пре сваке врсте написати редни број врсте, а изнад матрице ред са бројевима колона (видети тест пример).

---

★ Срећно на такмичењу! ★