

# Sadržaj

<b>1 Akademska godina 2005/06</b>	<b>4</b>
1.1 Test 1, 15.11.2005.	4
1.2 Test 2, prva grupa, 28.12.2005.	4
1.3 Test 2, druga grupa, 28.12.2005.	4
1.4 Programiranje 1, Završni ispit, januar, 25.01.2006	5
1.5 Programiranje I, Završni ispit, februar 2006	5
1.6 Programiranje I, Završni ispit, jun 2006.	5
1.7 Programiranje I, Završni ispit, septembar 2006.	6
1.8 Programiranje I, Završni ispit, oktobar 2006	6
<b>2 Akademska godina 2006/07</b>	<b>7</b>
2.1 Test 1, 28.11.2006., grupa 1	7
2.2 Test 1, 28.11.2006., grupa 2	7
2.3 Test 2, 06.02.2007., grupa 1	8
2.4 Test 2, 06.02.2007, grupa 2	8
2.5 Programiranje 1, Završni ispit, januar, 14.02.2007.	8
2.6 Programiranje I, Završni ispit, april 2007	8
2.7 Programiranje I, Završni ispit, septembar 2007	8
<b>3 Akademska godina 2007/08</b>	<b>10</b>
3.1 Programiranje I, prvi test, 28. novembar 2007.	10
3.2 Programiranje I, drugi test, 28. januar 2008.	10
3.3 Programiranje I, završni ispit, 28. januar 2008.	11
3.4 Programiranje I, završni ispit, 18. april 2008.	11
3.5 Programiranje I, završni ispit, septembar 2008.	12
<b>4 Akademska godina 2008/09</b>	<b>13</b>
4.1 Test 1, 28.11.2008.	13
4.2 Test 2, 17.01.2009.	14
4.3 Završni ispit, 05.02.2009.	14
4.4 Završni ispit, 15.04.2009.	15
4.5 Završni ispit, 29.08.2009.	15
4.6 Završni ispit, Oktobar 2, 2009	15
<b>5 Akademska godina 2009/2010</b>	<b>17</b>
5.1 Programiranje 1, prvi kolokvijum, 05.12.2009.	17
5.2 Programiranje 1, drugi kolokvijum	17
5.3 Programiranje 1, završni ispit	18
5.4 Programiranje 1, završni ispit, april 2010	18
5.5 Programiranje 1, završni ispit, septembar 2010	19

5.6	Programiranje 1, završni ispit, oktobar 2, 2010	20
<b>6</b>	<b>Akadska godina 2010/2011</b>	<b>21</b>
6.1	Programiranje 1, kolokvijum, 27.11.2010.	21
6.2	Programiranje 1, prvi mini-test na vežbama, 2010/2011	21
6.2.1	Grupa 1	21
6.2.2	Grupa 2	21
6.2.3	Grupa 3	22
6.2.4	Grupa 4	22
6.2.5	Grupa 5	22
6.3	Programiranje 1, drugi mini-test na vežbama, 2010/2011	22
6.3.1	Grupa 1	22
6.3.2	Grupa 2	22
6.3.3	Grupa 3	22
6.3.4	Grupa 4	23
6.3.5	Grupa 5	23
6.4	P1, završni ispit, januar 2011	23
6.5	P1, završni ispit, april 2011	23
<b>7</b>	<b>Akadska godina 2011/2012</b>	<b>25</b>
7.1	Programiranje 1 2011/2012, završni ispit, januar 2012	25
7.1.1	Grupa A	25
7.1.2	Grupa B	26
7.2	Programiranje 1 2011/2012, završni ispit, februar 2012	27
7.2.1	Deo II (radi se na papiru)	27
7.2.2	Deo II (radi se na računaru)	27
7.3	Programiranje 1 2011/2012, završni ispit, septembar 2012	29
7.3.1	Deo II (radi se na papiru)	29
7.3.2	Deo II (radi se na računaru)	29
7.4	Programiranje 1 2011/2012, završni ispit, oktobar 2012	30
<b>8</b>	<b>Akadska godina 2012/2013</b>	<b>33</b>
8.1	Test 4 - zadaci, 02.12.2012.	33
8.1.1	Grupa 1	33
8.1.2	Grupa 2	33
8.1.3	Grupa 3	33
8.2	I smer, Programiranje 1 2012/2013, završni ispit, februar 2013	35
8.3	I smer, Programiranje 1 2012/2013, završni ispit, septembar 2013	38
<b>9</b>	<b>Akadska godina 2013/2014</b>	<b>40</b>
9.1	Test 2, 07.11.2013	40
9.1.1	Grupa I	40
9.1.2	Grupa II	40
9.1.3	Grupa III	41
9.2	Test 4, 23.11.2013	41
9.2.1	Grupa I	41
9.2.2	Grupa II	42
9.2.3	Grupa III	43
9.3	Završni ispit, januarski rok, 23.01.2014	43
9.3.1	Deo 2, grupa 1	43
9.3.2	Deo 2, grupa 2	46
9.4	Završni ispit, februarski rok, 07.02.2014	48

9.5	Završni ispit, junski rok, 10.06.2014 . . . . .	50
9.5.1	URM . . . . .	50
9.5.2	Deo II . . . . .	50
9.6	Završni ispit, septembarski rok, 05.09.2014 . . . . .	52
<b>10</b>	<b>Akadska godina 2014/2015</b>	<b>55</b>
10.1	Test 2, 29.11.2014. . . . .	55
10.1.1	Grupa I . . . . .	55
10.1.2	Grupa II . . . . .	55
10.1.3	Grupa III . . . . .	56
10.2	Test 4, 13.01.2015. . . . .	56
10.2.1	Grupa I . . . . .	56
10.2.2	Grupa II . . . . .	57
10.2.3	Grupa III . . . . .	58
10.3	I smer, Programiranje 1 2014/2015, završni ispit, januar 2015, grupa I . . . . .	59
10.4	I smer, Programiranje 1 2014/2015, završni ispit, januar 2015, grupa II . . . . .	60
10.5	Programiranje 1, I smer, završni ispit, februar 2015. . . . .	62
10.6	I smer, Programiranje 1 2014/2015, završni ispit, jun 2015 . . . . .	63
<b>11</b>	<b>Akadska godina 2015/2016</b>	<b>66</b>
11.1	Programiranje 1, kolokvijum, 10.11.2016. . . . .	66
11.1.1	Grupa I . . . . .	66
11.1.2	Grupa II . . . . .	67
11.1.3	Grupa III . . . . .	67
11.2	Programiranje 1, Završni ispit, januar, 23.01.2016. . . . .	68
11.2.1	Grupa I . . . . .	68
11.2.2	Grupa II . . . . .	69
11.3	Programiranje 1, Završni ispit, februar, 12.02.2016. . . . .	71
11.4	1. Grupa, I smer, Programiranje 1 2015/2016, ispit, jun . . . . .	73
11.5	1. Grupa, I smer, Programiranje 1 2015/2016, ispit, septembar . . . . .	74
<b>12</b>	<b>Akadska godina 2016/2017</b>	<b>76</b>
12.1	Programiranje 1 - predlog kolokvijuma . . . . .	76
12.1.1	Grupa 1 . . . . .	76
12.1.2	Grupa 2 . . . . .	76
12.1.3	Grupa 3 . . . . .	77
12.2	Programiranje 1, Završni ispit, januar, 24.01.2017. . . . .	78
12.2.1	Grupa I . . . . .	78
12.2.2	Grupa 2 . . . . .	79
12.3	Programiranje 1, Završni ispit, februar, 02.02.2017. . . . .	81

# Glava 1

## Akademska godina 2005/06

### 1.1 Test 1, 15.11.2005.

1. Napisati program koji čita karaktere sa ulaza (dok ih ima) i sabira sve cifre koje se pojavljuju među njima (pretpostaviti da taj rezultat može biti smešten u promenljivu tipa `int`).

2. Napisati funkciju `f` sa argumentom `n` (celobrojnog tipa) čija je vrednost `n!` (faktorijel broja `n`) ako je `n` manje od 7, a inače je jednaka ostatku deljenja `n` sa 11. Napisati program u kojem se vrednost funkcije `f` štampa za vrednosti argumenta od 1 do `N`, gde je vrednost `N` učitana sa ulaza.

3. Napisati funkciju (sa prototipom `int palindrom(char s[])`) koja proverava da li je dati string `s` palindrom, tj. da se isto čita u oba smeru.

### 1.2 Test 2, prva grupa, 28.12.2005.

1. Odrediti dužinu najdužeg *uspona* u datom nizu `a`. Uspon od `i` do `j` u nizu `a` je opisan sa: za vrednosti `k1` i `k2` takve da je  $i \leq k1 < k2 \leq j$  važi `a[k1] < a[k2]`. Njegova dužina je tada `j-i+1`.

2. Sastaviti program koji iz komande linije učitava imena dve datoteke (ulazna i izlazna datoteka) i iz ulazne datoteke kopira u izlaznu svaki drugi karakter polazeći od prvog pročitanoog karaktera. U slučaju greške u otvaranju i zatvaranju datoteka, prijaviti odgovarajući komentar na `stdout` ili `stderr`.

3. Elementi niza  $F_i$  zadati su na sledeći način:

$$F_0 = 2$$

$$F_1 = 4$$

$$F_{n+2} = 2 \cdot F_{n+1} + 3 \cdot F_n, \text{ za } n \geq 0.$$

Napisati program koji za vrednost `k` zadatu sa standardnog ulaza izračunava i ispisuje vrednost  $F_k$ .

### 1.3 Test 2, druga grupa, 28.12.2005.

1. Odrediti dužinu najdužeg *platoa* u datom nizu `a`. Plato od `i` do `j` u nizu `a` je opisan sa: za vrednosti `k` takve da je  $i \leq k \leq j$  važi `a[i] = a[k]`. Njegova dužina je tada `j-i+1`.

2. Sastaviti program koji iz komande linije učitava imena dve datoteke (ulazna i izlazna datoteka) i iz ulazne datoteke kopira u izlaznu svaki drugi karakter polazeći od drugog pročitanoog karaktera. U slučaju greške u otvaranju i zatvaranju datoteka, prijaviti odgovarajući komentar na `stdin` ili `stderr`.

3. Elementi niza  $F_i$  zadati su na sledeći način:

$$F_0 = 1$$

$$F_1 = 3$$

$$F_n = 2 \cdot F_{n-1} + 3 \cdot F_{n-2}, \text{ za } n \geq 2.$$

Napisati program koji za vrednost `k` zadatu sa standardnog ulaza izračunava i ispisuje vrednost  $F_k$ .

## 1.4 Programiranje 1, Završni ispit, januar, 25.01.2006

1. Sastaviti funkciju koja za data dva stringa ispituje da li su jednaka (bez korišćenja funkcije `strcmp`).
2. Sastaviti funkciju koji učitava datum u obliku `dd mm gggg` i izračunava datum sledećeg dana (pretpostaviti da svaki februar ima 28 dana).
3. Sastaviti funkciju koji kao parametar dobija vrednost celog broja koji predstavlja kursor (u dinarima), a vraća broj novčanica i metalnih novčića (u maksimalnim apoenima) koji su potrebni da se kursor vrati. Npr.

$$97 = (0 \times 1) + (1 \times 2) + (1 \times 5) + (2 \times 20) + (1 \times 50) + (0 \times 100) + (0 \times 500) + (0 \times 1000)$$

4. Sastaviti program koji prima kao argument ime datoteke koju treba otvoriti. Tu datoteku treba pročitati i ispisati (na standardnom izlazu) koja cifra (među svim ciframa koje se pojavljuju u datoteci) ima najveći broj pojavljivanja.

## 1.5 Programiranje I, Završni ispit, februar 2006

1.  $n$ -tocifreni broj je Armstrongov, ako je jednak sumi  $n$ -tih stepena svojih cifara. Napisati program koji za uneti broj ispituje da li je Armstrongov.

2. Napisati funkciju `int substring(char* s1, char* s2)` čiji su argumenti niske `s1` i `s2` i koja vraća vrednost 1 ako je `s2` podniska niske `s1`, a 0 inače.

3. Elementi niza  $F_i$  zadati su na sledeći način:

$$F_0 = a$$

$$F_1 = b$$

$$F_{n+2} = F_{n+1} + F_n, \text{ za } n \geq 0.$$

- Napisati program koji za celobrojne vrednosti  $a$ ,  $b$  i  $k$  zadate sa standardnog ulaza izračunava i ispisuje vrednost  $F_k$ . U programu ne koristiti nizove.

4. Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj, a na standardni izlaz ispisuje vrednost tog broja sa razmenjenim vrednostima bitova na pozicijama  $i$ ,  $j$ . Pozicije  $i$ ,  $j$  se učitavaju kao parametri komandne linije. Smatrati da je krajnji desni bit binarne reprezentacije 0-ti bit. Pri rešavanju nije dozvoljeno koristiti pomoćni niz niti aritmetičke operatore  $+$ ,  $-$ ,  $/$ ,  $*$ ,  $\%$ .

## 1.6 Programiranje I, Završni ispit, jun 2006.

1. Niz  $T(n)$  dat je sledećom vezom:

$$T(n+3) = a \cdot T(n+2) + b \cdot T(n+1) + c \cdot T(n)$$

i uslovima  $T(0) = d$ ,  $T(1) = e$ ,  $T(2) = f$ . Napisati program koji sa standardnog izlaza čita realne vrednosti  $a, b, c, d, e, f, n$  i na standardni izlaz ispisuje vrednost  $T(n)$ . Program ne sme da koristi rekurziju, niti nizove.

2. Napisati program koji prima ceo broj  $n$  kao argument komandne linije i ispisuje razliku broja jedinica na parnim i neparnim pozicijama u binarnom zapisu broja  $n$ .

PRIMER: za  $n = 19 = (10011)_2$  izlaz je 1.

3. Prvi red standardne ulazne datoteke sadrži 2 cela broja manja od 50 koji predstavljaju redom broj vrsta i broj kolona realne matrice  $A$ . Svaki sledeći red sadrži po jednu vrstu matrice. Napisati program koji pronalazi sve elemente matrice  $A$  koji su jednaki zbiru svih svojih susednih elemenata i štampa ih u obliku (broj vrste, broj kolone, vrednost elementa).

## 1.7 Programiranje I, Završni ispit, septembar 2006.

1. Napisati program koji sledeću sumu računa sa minimalnim brojem operacija

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!}$$

Celobrojne vrednosti brojeva  $n$  i  $x$  se učitavaju sa standardnog ulaza.

2. Napisati funkciju koja prima jedan argument  $n$  tipa `int` i koja vraća razliku broja jedinica na parnim i neparnim pozicijama u binarnom zapisu argumenta.

PRIMER: za  $n = 19 = (10011)_2$  funkcija vraća 1.

Napisati program koji pozivom ove funkcije računa i štampa razliku broja jedinica za sve argumente komandne linije (pretpostavka je da su argumenti komandne linije zadati korektno).

3. (a) Napisati funkciju `void brojanje(int a[], int brojac[], int N)` čiji su argumenti  $a$  i `brojac` celobrojni nizovi dimenzije  $N$ . Vrednosti elemenata niza  $a$  su između 0 i  $N-1$ . Funkcija izračunava elemente niza `brojac` tako da je  $i$ -ti element `brojac[i]` jednak broju pojavljivanja broja  $i$  u nizu  $a$ .  
(b) Za celobrojni niz  $a$  dimenzije  $N$  (opisan u prethodnom pasusu) kažemo da je *permutacija* ako sadrži sve brojeve  $i$ :  $0 \leq i \leq N$ . Sastaviti funkciju `int DaLiJePermutacija(int a[], int N)` koja vraća 1 ako je niz  $a$  permutacija, a 0 inače. (koristiti funkciju `brojanje`).

## 1.8 Programiranje I, Završni ispit, oktobar 2006

1. (a) Napisati funkciju koja ispituje da li su dve niske (koje se prenose kao parametri funkcije) anagrami. Anagrami su niske koje se sastoje od istih karaktera. Npr. `vetar`, `trave`, `verat` su anagrami.

- (b) Napisati program koji testira funkciju iz prvog dela.

2. Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj, a na standardni izlaz ispisuje vrednost tog broja sa razmenjenim vrednostima bitova na poziciji  $i$ ,  $j$ . Pozicije  $i$ ,  $j$  se učitavaju kao parametri komandne linije. Smatrati da krajnji desni bit binarne reprezentacije je 0-ti bit. Pri rešavanju nije dozvoljeno koristiti pomoćni niz niti aritmetičke operatore `+`, `-`, `/`, `*`, `%`.

3. Prvi red datoteke "ulaz.txt" sadrži 2 cela broja manja od 50 koji predstavljaju redom broj vrsta i broj kolona realne matrice  $A$ . Svaki sledeći red sadrži po jednu vrstu matrice. Napisati program koji nalazi i štampa sve četvorke oblika  $(A(i, j), A(i+1, j), A(i, j+1), A(i+1, j+1))$  u kojima su svi elementi međusobno različiti.

## Glava 2

# Akademska godina 2006/07

### 2.1 Test 1, 28.11.2006., grupa 1

1.

(a) Opisati BNF notacijom i sintaksnim dijagramom kategoriju dekadnog realnog broja koji ima ne više od dve decimale.

(b) Ako je data gramatika  $G = (\{S, A, B\}, \{a, b, c\}, P, S)$  sa pravilima iz skupa  $P$ :

$$S \rightarrow AcB$$
$$A \rightarrow aB|c$$
$$B \rightarrow bA|c$$

odrediti koje od sledećih niski pripadaju skupu  $L(G)$ : (i)  $abacbab$ ; (ii)  $acb$ ; (iii)  $ccc$ .

2. Napisati C program koji učitava sa standardnog ulaza cele brojeve dok ih je manje od 100 ili dok ne naiđe na nulu. Program treba da ispiše na standardni izlaz minimum unetih brojeva i koliko puta se taj minimum pojavio među unetim brojevima (ne koristiti nizove).

3. Napisati C program koji na standardni izlaz ispisuje odgovor da li je uneti prirodan broj  $x$  deljiv svim svojim ciframa (na primer, 105 nije, 248 jeste).

### 2.2 Test 1, 28.11.2006., grupa 2

1.

(a) Opisati BNF notacijom i sintaksnim dijagramom kategoriju označenog celog dekadnog broja koji ima ne više od tri cifre.

(b) Ako je data gramatika  $G = (S, A, B, a, b, c, P, S)$  sa pravilima iz skupa  $P$ :

$$S \rightarrow AcB$$
$$A \rightarrow aB|c$$
$$B \rightarrow bA|c$$

odrediti koje od sledećih niski pripadaju skupu  $L(G)$ : (i)  $ccbc$ ; (ii)  $cbcba$ ; (iii)  $accbac$ .

2. Napisati C program koji učitava sa standardnog ulaza cele brojeve dok ih je manje od 100 ili dok ne naiđe na nulu. Program treba da ispiše na standardni izlaz broj sa minimalnom poslednjom cifrom. Ako ih ima više neka ispiše bilo koji (ne koristiti nizove).

3. Napisati C program koji na standardni izlaz ispisuje odgovor da li je uneti prirodan broj deljiv sumom svojih cifara.

## 2.3 Test 2, 06.02.2007., grupa 1

1. Napisati funkciju koja proverava da li je dati broj palindrom ako je on zadat kao niska karaktera.
2. Napisati C program koji učitava sa standardnog ulaza datum u formatu `dd.mm.gggg` i ispisuje na standardni izlaz datum prethodnog dana.
3. Napisati C program koji iz komandne linije učitava nazive nekoliko datoteka i ispisuje na standardni izlaz, za svaku datoteku posebno, dekadnu cifru koje se najčešće pojavljuje u toj datoteci. Ako ima više cifara sa istim najvećim brojem pojava, ispisati najmanju. U slučaju greške u otvaranju datoteke, prijaviti odgovarajući komentar na `stderr`.

## 2.4 Test 2, 06.02.2007, grupa 2

1. Napisati funkciju koja proverava da li je dati broj palindrom ako je on zadat kao ceo broj.
2. Napisati C program koji učitava sa standardnog ulaza datum u formatu `dd.mm.gggg` i ispisuje na standardni izlaz datum sledećeg dana.
3. Napisati C program koji iz komandne linije učitava naziv datoteke i ispisuje na standardni izlaz slovo engleske abecede koje se najčešće pojavljuje u datoteci. Ako ima više slova sa istim najvećim brojem pojava, ispisati leksikografski najmanje slovo. U slučaju greške u otvaranju datoteke, prijaviti odgovarajući komentar na `stderr`.

## 2.5 Programiranje 1, Završni ispit, januar, 14.02.2007.

1. Napisati funkciju `int fibonaci(int n)`; koja računa n-ti element Fibonačijevog niza ( $n \leq 50$ ). (Fibonačijev niz dat je uslovima  $F_0 = 1, F_1 = 1, F_{n+2} = F_{n+1} + F_n$ .)
2. Napraviti program koji učitava ime datoteke iz komandne linije i zatim određuje maksimalni ceo broj koji se pojavljuje u toj datoteci (u rešenju ne koristiti nizove). Pretpostavljamo da datoteka sadrži samo cele brojeve. Ukoliko u komandnoj liniji nije navedeno ime datoteke, poruku o tome ispisati na `stderr`.
3. Napisati funkciju koja izračunava sumu bitova celog broja `x`.
4. Napisati program koji sa standardnog ulaza učitava dva stringa. Proveriti da li se svi karakteri prvog stringa nalaze u drugom stringu. Na kraju ispisati odgovarajuću poruku (npr. `aka` se nalazi u `bajka`, `baba` se nalazi u `bajka`, `majka` se ne nalazi u `bajka`.)

## 2.6 Programiranje I, Završni ispit, april 2007

1. Napisati program koji iz datoteke koja se zadaje kao prvi argument komandne linije kopira svaki drugi karakter u datoteku koja se zadaje kao drugi argument komandne linije.
2. Napisati program koji sa standardnog ulaza učitava 50 celih brojeva i razdvaja ih na parne i neparne tako što parne brojeve upisuje na početak niza, a neparne na kraj niza. Ispisati niz dobijen na taj način. Nije dozvoljeno koristiti dodatne nizove.
3. Napisati funkciju `int podniska(char* s1, char* s2)` koja proverava da li je niska `s1` podniska niske `s2`. (vraća 1 ako jeste podniska, 0 ako nije).

## 2.7 Programiranje I, Završni ispit, septembar 2007

1. Napisati funkciju `int podstring(char s1[], char s2[])` koja proverava da li je prvi string podstring drugog i vraća 1 ako jeste odnosno 0 ako nije.
2. Napisati program koji iz datoteke čije se ime unosi sa standardnog ulaza čita tekst i prepisuje ga na standardni izlaz s tim da sve višestruke razmake u tekstu zamenjuje jednostrukim razmakom. Npr, za uneto:



"Zdravo svete"

program treba da ispiše:

"Zdravo svete"

3. Napisati program koji za ceo broj koji se unosi sa standardnog ulaza proverava da li je deljiv sumom svojih cifara i ispisuje poruku o tome na standardni izlaz.
4. Napisati program koji sledeću sumu računa sa što manjim brojem operacija:

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n * \frac{x^{2n}}{(2n)!}$$

Realan broj  $x$  i ceo broj  $n$  se unose sa standardnog ulaza.

## Glava 3

# Akademska godina 2007/08

### 3.1 Programiranje I, prvi test, 28. novembar 2007.

1. Neka je data gramatika  $G = (\Sigma, N, S, P)$  gde je  $\Sigma = \{a, b\}$   $P = \{S \rightarrow Sb \ (1^\circ), S \rightarrow a(2^\circ)\}$ .
  - (a) Dokazati da reč `abbb` može da se izvede u toj gramatici.
  - (b) Zapisati pravila ove gramatike redom u: BNF-u, EBNF-u i preko sintasnih dijagrama.
2. Napisati URM program koji izračunava funkciju:

$$f(x, y) = \begin{cases} 0 & , \text{ ako } x \geq y \\ 1 & , \text{ inače} \end{cases}$$

3. Napisati C program koji sa standardnog ulaza učitava ceo broj  $n$  i zatim proverava da li je  $n$  veće od 10. Ako jeste, program ispisuje na standardni izlaz ostatak pri deljenju broja  $n$  sa 3, a ako nije program ispisuje na standardni izlaz ceo deo pri deljenju broja  $n$  sa 3.
4. Napisati funkciju u C-u `int faktorijel(int n)` koja izračunava faktorijel celog broja  $n$  i program koji tu funkciju poziva (program od korisnika traži da unese ceo broj i onda štampa vrednost faktorijala). Na primer,

```
Unesite ceo broj: 3
Faktorijal broja 3 je: 6
```

### 3.2 Programiranje I, drugi test, 28. januar 2008.

1. Napisati program koji sa standardnog ulaza učitava karaktersku nisku (dužine ne veće od 9 karaktera) i karakter i nakon toga pozivom funkcije `int poslednja_pojava(char s[], char c)` određuje i ispisuje poslednju pojavu datog karaktera u datoj nisci. Takođe napisati i funkciju `int poslednja_pojava(char s[], char c)` koja vraća  $-1$  ako se karakter  $c$  ne nalazi u nisci  $s$  odnosno vraća poziciju na kojoj se taj karakter nalazi u nisci.
2. Napisati program koji iz datoteke `ulaz.txt` prepisuje u datoteku `izlaz.txt` svaki drugi karakter.
3. Napisati program koji za uneti ceo broj  $n$  sa standardnog ulaza proverava da li je deljiv sumom svojih cifara.

### 3.3 Programiranje I, završni ispit, 28. januar 2008.

1. Napisati program koji sa standardnog ulaza učitava ne više od 50 celih brojeva, određuje i ispisuje na standardni izlaz maksimalni element niza i koliko puta se on pojavio u nizu.

Na primer, ako je unet sledeći niz:

```
15 2 8 3 13 15 2 15
```

program treba da ispiše da je maksimalni element niza broj 15 i da se on pojavio u nizu 3 puta.

2. Napisati program koji sa standardnog ulaza učitava neoznačen ceo broj  $n$  i zatim ispisuje na standardni izlaz sumu bitova binarne reprezentacije tog celog broja.

Na primer, za uneto  $n = 5$  program treba da ispiše 2 (pošto je binarni zapis broja 5: 101), za uneto  $n = 11$  treba da ispiše 3 (pošto je binarni zapis broja 11: 1011).

3. Napisati program koji u što manjem broju koraka računa sledeću sumu:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n * \frac{x^{2n+1}}{(2n+1)!}$$

Ceo broj  $n$  i realan broj  $x$  se unose sa standardnog ulaza.

4. Napisati program koji iz datoteke čije se ime zadaje kao prvi argument komadne linije učitava podatke o studentima. U svakom redu datoteke se nalaze redom: ime studenta (karakterska niska dužine ne veće od 15 karaktera), prezime studenta (karakterska niska dužine ne veće od 20 karaktera), broj indeksa studenta (ceo broj) i prosek ocena (realan broj). Pretpostavka je da se u datoteci nalaze podaci za maksimalno 100 studenata.

Prvo formirati niz struktura studenata koji treba popuniti čitanjem iz datoteke pa onda nakon toga ispisati podatke o studentima na standardni izlaz (čitajući ovaj put podatke iz niza).

### 3.4 Programiranje I, završni ispit, 18. april 2008.

1. Napisati program koji učitava sa standardnog ulaza cele brojeve dok se ne unese nula kao oznaka za kraj. Program treba da ispiše na standardni izlaz minimum unetih brojeva i koliko puta se taj minimum pojavio među unetim brojevima (ne koristiti nizove).
2. Napisati program koji na standardni izlaz ispisuje odgovor da li je uneti prirodan broj  $x$  deljiv svim svojim ciframa (na primer: 105 i 23 nisu a 12 i 248 jesu deljivi svim svojim ciframa).
3. Napisati funkciju `int pozicija(char s[], char c)` koja proverava da li se karakter  $c$  nalazi u okviru niske  $s$  (dužine maksimalno 20) i vraća  $-1$  ako se karakter ne nalazi u nisci odnosno odgovarajuću poziciju (počev od početka niske) na kojoj se karakter nalazi. Nakon toga napisati program koji nisku  $s$  i karakter  $c$  unosi sa standardnog ulaza i na standardni izlaz štampa odgovarajuću poruku.

Primer:

```
Niska: abcda Karakter: e
```

```
Karakter e se ne nalazi Karakter a se nalazi na poziciji 0
```

```
Karakter b se nalazi na poziciji 1
```

4. Napisati program koji prima kao argument komandne linije ime datoteke iz koje treba pročitati i ispisati (na standardni izlaz) koja cifra (među svim ciframa koje se pojavljuju u datoteci) ima najveći broj pojavljivanja.

### 3.5 Programiranje I, završni ispit, septembar 2008.

1. Elementi niza  $F_i$  zadati su na sledeći način:

$$F_0 = 1$$

$$F_1 = 3$$

$$F_n = 2 \cdot F_{n-1} + 3 \cdot F_{n-2}, \text{ za } n \geq 2.$$

Napisati program koji za vrednost  $k$  zadatu sa standardnog ulaza izračunava i ispisuje vrednost  $F_k$ .

Napomena: program ne sme da koristi rekurziju, niti nizove.

2. Napisati program koji sa standardnog ulaza učitava ne više od 50 celih brojeva, određuje i ispisuje na standardni izlaz maksimalni element niza. Oznaka za kraj unosa je uneta nula (ili unetih 50 celih brojeva).

Na primer, ako je unet sledeći niz:

15 2 8 3 13 15 2 15

program treba da ispiše da je maksimalni element niza broj 15.

3. Napisati program koji za ceo broj koji se unosi sa standardnog ulaza proverava da li je deljiv sumom svojih cifara i ispisuje poruku o tome na standardni izlaz.

4. Sastaviti program koji iz komande linije učitava ime datoteke i reč na engleskom jeziku. U datoteci se nalazi spisak engleskih reči i njihov prevod na srpski jezik (u svakom redu po jedan par reči). Na standardni izlaz ispisati odgovarajući prevod date reči ili podatak o tome da se reč ne nalazi u datoteci. Reči u datoteci su dučine manje od 20.

Na primer, za datoteku:

```
apple jabuka  
pineapple ananas  
orange narandza  
pear kruska  
grape grozdje
```

i reč *orange* program treba da ispiše *narandza* a za reč *cherry* program treba da ispiše poruku *Rec se ne nalazi u rečniku*. U programu se mogu koristiti funkcije iz zaglavlja *string.h*.

# Glava 4

## Akademska godina 2008/09

### 4.1 Test 1, 28.11.2008.

1. Napisati URM program koji izračunava funkciju

$$f(x, y) = \begin{cases} x & , \text{ ako } x \leq y \\ y & , \text{ inače} \end{cases}$$

2. Napisati funkciju `int sumafaktorijala(int x, int y)` koja za dva pozitivna cela broja  $x$  i  $y$  računa sumu faktorijala  $x! + y!$  i program koji sa standardnog ulaza učitava pozitivne brojeve  $x$  i  $y$ , pozivom funkcije `sumafaktorijala` izračunava sumu  $x! + y!$  i ispisuje je na standardni izlaz (izlaz iz programa treba da izgleda kao u primeru).

Primer:

```
Unesite dva cela broja:
```

```
4
```

```
3
```

```
Suma faktorijala brojeva 4 i 3 je: 30
```

3. Napisati program koji sa standardnog ulaza učitava prvo pozitivan ceo broj  $n$ , a zatim i  $n$  celih brojeva i izračunava zbir parnih. Izračunati zbir ispisati na standardni izlaz (izlaz iz programa treba da izgleda kao u primeru).

Primer:

```
Unesite ceo broj: 5
```

```
Unesite 5 brojeva:
```

```
2
```

```
12
```

```
35
```

```
11
```

```
8
```

```
Zbir parnih je: 22
```

Tačan odgovor na pitanje od 1 do 10 (deo I) nosi 5 bodova. Odgovor „(e) ne znam“ nosi 0 poena, a pogrešan odgovor -1 poen. Zadaci (deo II) nose 50 bodova.

## 4.2 Test 2, 17.01.2009.

1. Napisati funkciju `kopiraj_n(char* t, char* s, int n)` koja iz karakterske niske `s` kopira najviše prvih `n` karaktera u karaktersku nisku `t` i program koji je poziva. Terminirajuća nula iz niske `s` se kopira u nisku `t` samo ako je `n` veće od dužine niske `s`.
2. Napisati program koji sa standardnog ulaza učitava vrednost `c` tipa `unsigned char` i ceo broj `n` i zatim ispisuje rezultat poziva funkcije `unsigned char rotiraj(unsigned char c, int n)`. Napisati funkciju `rotiraj` koja rotira binarni zapis `c` za `n` pozicija u desno.
3. Napisati program koji za dve datoteke čija su imena data kao prvi i drugi argument komandne linije, radi sledeće: za cifru u prvoj datoteci, u drugu datoteku se upisuje 0, za slovo se upisuje 1, a za sve ostale karaktere se upisuje 2.

## 4.3 Završni ispit, 05.02.2009.

1. Data je datoteka `recnik.txt` koja u svakom redu ima po dve reči: reč na srpskom jeziku i njen prevod na engleski jezik. Reči su dužine najviše 20 karaktera. Napisati program koji za zadatu reč sa standardnog ulaza, na osnovu sadržaja datoteke `recnik.txt`, proverava da li postoji njen prevod u rečniku. Ako postoji, ispisati ga, a ako ne postoji ispisati odgovarajuću poruku.

Na primer, ako je sadržaj datoteke:

```
bioskop cinema
slika picture
priroda nature
```

Program treba da izgleda ovako:

```
Unesite rec na srpskom: slika
Prevod na engleski je: picture.
```

2. Napisati program koji za argumente komandne linije sa kojima je pozvan proverava da li postoje dva ista argumenta i ispisuje na standardni izlaz odgovarajuću poruku kao odgovor.

Primer rada programa:

```
./zadatak2 jelena marko danica petar nikola danica djordje
Postoje dva ista argumenta.
```

3. Napisati program koji sa standardnog ulaza čitava najpre broj artikala (ceo broj manji od 20) a zatim podatke o artiklima. Artikli su voćke koje imaju po dva podatka: naziv voćke i cenu (naziv voćke je karakterska niska dužine do 20 karaktera). Program potom traži od korisnika da unese neku cenu i štampa na standardni izlaz sve voćke koje imaju zadatu cenu.

Primer rada programa:

```
4
jabuka 30
kruska 40
```

ananas 60  
limun 40

Unesite cenu: 40  
Voce te cene je: kruska limun

4. Aritmetička sredina niza je zbir svih elemenata niza podeljen brojem elemenata niza. Napisati program koji sa standardnog ulaza unosi broj elemenata niza (ne više od 20) pa zatim i elemente niza i pronalazi koliko elemenata niza je veće od aritmetičke sredine niza i štampa taj podatak na standardni izlaz.  
Za niz: 3.4 6 14.6 6 0 21, aritmetička sredina je 8.5, pa odgovor treba da bude: 2

Napomena: Mogu se koristiti funkcije iz zaglavlja *string.h*, ukoliko je potrebno.

#### 4.4 Završni ispit, 15.04.2009.

- Napisati funkciju koja računa sumu bitova neoznačenog broja i program koji je poziva.
- Napisati program koji dobija dve datoteke kao argumente komandne linije i iz prve datoteke upisuje u drugu karakter C ako je pročitana cifra u prvoj datoteci, karakter S ako je pročitano slovo i karakter O za ostalo.
- Napisati funkciju `int palindrom(char s[])` koja proverava da li data karakterska niska predstavlja palindrom (da li se isto čita sa obe strane) i program koji je poziva. Program sa standardnog ulaza treba da učitava nisku (dužine ne veće od 20) i nakon toga da ispiše na standardni izlaz da li je uneta niska palindrom ili ne.
- Napisati program koji sa standardnog ulaza čita prvo broj voćki (manji od 20) pa zatim podatke o vočkama, ime voćke i cenu voćke. Nakon toga program od korisnika traži da unese ime voćke i štampa cenu te voćke na standardnom izlazu ako takva voćka postoji odnosno odgovarajuću poruku ako ne postoji.

#### 4.5 Završni ispit, 29.08.2009.

1. Napisati program koji iz datoteke čije je ime dato kao prvi argument komandne linije prepisuje u datoteku čije je ime dato kao drugi argument komandne linije samo one reči čija je dužina veća od 10. Pretpostavlja se da su argumenti komande linije pravilno zadati.
2. Napisati program koji sa standardnog ulaza učitava prvo ceo broj  $n$  pa zatim  $n$  celih brojeva. Odrediti broj elemenata niza koji su deljivi srednjom vrednošću niza i sve ih ispisati.
3. Niz  $T_n$  zadat je na sledeći način:  $T_0 = 1$ ,  $T_1 = 2$ ,  $T_n = 2 \cdot T_{n-1} + 3 \cdot T_{n-2}$ . Napisati iterativnu i rekursivnu verziju funkcije `int t(int n)` koja za zadato  $n$  računa  $T_n$ . Napisati program koji za broj  $n$  zadat sa standardnog ulaza ispisuje vrednost  $T_n$ .

#### 4.6 Završni ispit, Oktobar 2, 2009

1.  $n$ -tocifreni broj je Armstrongov, ako je jednak sumi  $n$ -tih stepena svojih cifara. Napisati program koji za uneti ceo broj ispituje da li je Armstrongov.

2. Napisati funkciju `int substring(char* s1, char* s2)` čiji su argumenti niske `s1` i `s2` i koja vraća vrednost 1 ako je `s2` podniska niske `s1`, a 0 inače.

3. Elementi niza  $F_i$  zadati su na sledeći način:

$$F_0 = a$$

$$F_1 = b$$

$$F_{n+2} = F_{n+1} + F_n, \text{ za } n \geq 0.$$

Napisati program koji za celobrojne vrednosti `a`, `b` i `k` zadate sa standardnog ulaza izračunava i ispisuje vrednost  $F_k$ . U programu ne koristiti nizove.

4. Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj, `a` na standardni izlaz ispisuje vrednost tog broja sa razmenjenim vrednostima bitova na pozicijama `i`, `j`. Pozicije `i`, `j` se učitavaju kao parametri komandne linije. Smatrati da je krajnji desni bit binarne reprezentacije 0-ti bit. Pri rešavanju nije dozvoljeno koristiti pomoćni niz niti aritmetičke operatore `+`, `-`, `/`, `*`, `%`.



## Glava 5

# Akademski godina 2009/2010

### 5.1 Programiranje 1, prvi kolokvijum, 05.12.2009.

1. Napisati URM program koji izračunava funkciju  $f(x, y) = 2x + y$ .

2. Napisati funkciju sa prototipom

```
int razmeni(int);
```

koja razmenjuje cifru najveće i cifru najmanje težine zadatog celog broja (između 1 i 999) i vraća tako dobijeni broj. Na primer, za ulaz 128 izlaz funkcija vraća 821, za ulaz 56 funkcija vraća 65, za ulaz 9 funkcija vraća 9.

Napisati i program koji poziva funkciju `razmeni` za sve brojeve od 1 do 100 i ispisuje njene rezultate.

3. Napisati program koji za uneti datum u obliku: 23 10 2008 (dan, mesec pa godina) određuje i štampa (u istom obliku) datum prethodnog dana (voditi računa o prestupnim godinama).

### 5.2 Programiranje 1, drugi kolokvijum

1. Definisati:

- tip strukture `kompleksan` koja sadrži dva člana tipa `float` koja predstavljaju realni i imaginarni deo kompleksnog broja;
- funkciju koja ispisuje kompleksan broj u obliku  $\langle \text{realni deo} \rangle + i * \langle \text{imaginarni deo} \rangle$ ;
- funkciju koja množi dva kompleksna broja i vraća kompleksan broj kao rezultat.

Napisati program koji učitava dva kompleksna broja sa standardnog ulaza, izračunava i štampa njihov proizvod.

2. Napisati funkciju koja sa standardnog ulaza učitava u zadati niz najviše  $n$  celih brojeva (gde je  $n$  takođe argument funkcije) i vraća broj uspešno učitanih.

Napisati funkciju koja računa i vraća prosek prvih  $n$  elemenata zadatog niza celih brojeva (gde je  $n$  takođe argument funkcije).

Napisati program koji korišćenjem prethodnih funkcija učitava niz celih brojeva (najviše 100) i ispisuje elemente koji su veći od proseka niza, kao i ukupan broj takvih elemenata.

3. Niz  $T(n)$  zadat je na sledeći način:  $T(0) = 3$ ,  $T(1) = 1$ ,  $T(n) = 4 * T(n - 1) - 2 * T(n - 2)$ ,  $n \geq 2$ . Napisati iterativnu i rekurzivnu verziju funkcije koja za zadatu vrednost  $n$  računa  $T(n)$ . Napisati program koji za broj  $n$  zadat sa standardnog ulaza ispisuje vrednost  $T(n)$  pozivom i jedne i druge funkcije.

## 5.3 Programiranje 1, završni ispit

1. Napisati sledece funkcije:

```
int poredjenje(char* s1, char* s2);
// vraca 1 ako su s1 i s2 iste niske, 0 u suprotnom

void uVelikaSlova(char* s);
// pretvara sva slova niske s u velika, ostale znakove ne menja
```

Napisati program koji sa standardnog ulaza učitava dve reči (dužine najviše 20 znakova) i, koristeći ove dve funkcije, ispisuje da li su one jednake ako se sva slova pretvore u velika slova. Na primer, reči "isPit2010" i "IsPiT2010" su jednake ako se sva slova pretvore u velika.

2. Napisati program koji za zadato  $n$  ( $n \geq 1$ ) iscrtava figuru u vidu slova X. Primer za  $n = 2$ :

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

Primer za  $n = 3$ :

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

3. Ime tekstualne datoteke je parametar komandne linije. Datoteka sadži ime, prezime studenta i niz ocena (tačno 10 ocena za svakog studenta, ocene idu od 5 do 10), najviše za 100 studenata. Program treba da ispisuje ime, prezime i prosek studenta sa najvećim prosekom. Ako ima više takvih studenata, ispisati podatke za sve. Koristiti strukturu:

```
typedef struct
{
    char ime[20];
    char prezime[20];
    int ocene[10];
} STUDENT;
```

4. Ako je data tekstualna datoteka "plain.txt" napraviti tekstualnu datoteku "sifra.txt" tako što se svako slovo zamenjuje svojim prethodnikom (ciklično) suprotne velicine 'b' sa 'A', 'B' sa 'a', 'a' sa 'Z', 'A' sa 'z', itd. Podrazumevati da se na sistemu koristi tabela karaktera ASCII.

## 5.4 Programiranje 1, završni ispit, april 2010

1. Napisati funkciju koja vrši prevođenje niske znakova koja predstavlja broj u proizvoljnoj osnovi u ceo broj (int). Osnova je data kao broj između 2 i 36 (uključujući granice), a moguće cifre su znaci '0', '1', ... Za osnove veće od 10, dodatne cifre su mala ili velika slova engleskog alfabeta (nema razlike između

malih i velikih slova), pri čemu važi da 'a' ima vrednost 10, 'b' vrednost 11 i tako redom. Pretpostaviti da rezultat može da stane u `int`.

Prototip funkcije je:

```
int uInt(char *cifre, int n, int osnova);
```

Napisati prateći program koji omogućava korisniku da unese prvo osnovu u kojoj radi, zatim broj cifara broja koji želi da unese (maksimalno 10) a potom i same cifre.

Primer: za osnovu 12,  $n = 4$  i  $cifre = \{ 'b', 'B', '1', '0' \}$  rezultat treba da bude  $11 * 12^3 + 11 * 12^2 + 1 * 12^1 + 0 * 12^0 = 20604$ .

2. Napisati funkciju: `int palindrom(char *s)` koja proverava da li je data niska palindrom i vraća 1 ako jeste, odnosno 0 ako nije. Nakon toga napisati program koji iz datoteke čije se ime zadaje kao prvi argument komandne linije čita sve reči i na standardni izlaz ispisuje samo one reči koje su palindromi. Palindrom je ona niska koja se isto čita sa leve i sa desne strane.
3. Definisati strukturu koja opisuje automobil atributima marka i cena (marka je karakterska niska dužine ne više od 20 karaktera, a cena je ceo broj). Data je datoteka "automobili.txt" koja na početku sadrži broj automobila koji su u njoj opisani, a potom redom podatke o automobilima. Uz pretpostavku da datoteka sadrži podatke o najviše 100 automobila, napisati program koji računa prosečnu cenu automobila i ispisuje je, a potom ispisuje marku automobila čija cena se najviše razlikuje od dobijenog proseka (ako ima više automobila sa istom razlikom, ispisati prvi). Pretpostavka je da je datoteka ispravno formatirana.

## 5.5 Programiranje 1, završni ispit, septembar 2010

1. Poznato je da je razvoj eksponencijalne funkcije  $e^x$  u okolini tačke  $x_0 = 0$  dat sumom:

$$S = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

Parcijalne sume koje sadrže samo prvih  $k$  sabiraka ove sume definišu se kao

$$S_k = \sum_{i=0}^k \frac{x^i}{i!}$$

Napisati funkciju `double eksp(double x)` koja računa  $e^x$  ( $x \geq 0$ ) koristeći gore navedenu sumu. Računanje treba da prestane kada razlika dve uzastopne parcijalne sume bude manja od  $10^{-3}$ . Napisati prateći program koji sa tastature učitava  $x$  i ispisuje  $e^x$  na 3 decimale koristeći funkciju `eksp`. NAPOMENA: voditi računa o potencijalnom prekoračenju brojioca i/ili imenioca.

2. a) Napisati funkciju `void obrni(char rec[], int k)` koja rotira reč za  $k$  mesta ulevo, napr. za reč `sveska` i  $k=2$ , rezultat treba da bude `eskasv`.  
b) Napisati program koji ima dva argumenta — ime tekstualne datoteke i prirodan broj  $k$ . Podrazumeva se da zadata datoteka sadrži samo slova i beline i da je svaka reč iz datoteke dužine najviše 100. Program treba da učitava reči iz datoteke, da svaku reč rotira za  $k$  mesta i da tako dobijenu reč upiše u datoteku čije je ime `rotirano.txt`.
3. Data je datoteka `obdaniste.txt`. U svakom redu datoteke (ima ih najviše 100), nalazi se ime deteta (ne vece od 20 karaktera), pol deteta (m ili z) i ocena. Ocenu je svako dete dalo radu obdaništa. Pretpostavlja se da je datoteka dobro formatirana. Napisati program koji:
  - a) učitava podatke iz datoteke u niz, dok ne dodje do kraja datoteke. Koristiti strukturu:

```

typedef struct
{
    char ime[20];
    char pol;
    int ocena;
} DEFE;

```

- b) ispisuje u datoteku `statistika.txt` koliko ima dečaka, a koliko devojčica i prosečnu ocenu. Potom ispisuje imena dece brojnijeg pola.

## 5.6 Programiranje 1, završni ispit, oktobar 2, 2010

1. Date su dve datoteke (imena datoteka su argumenti komadne linije). U prvoj datoteci u svakom redu nalazi se ime proizvoda (ne više od 20 karaktera) i količina u kojoj se proizvodi (najviše 100 redova, proizvodi se mogu ponavljati), a u drugoj datoteci nalazi se ime proizvoda i količina u kojoj se potražuje (najviše 100 redova, proizvodi se ne ponavljaju). Potrebno je pronaći proizvod koji se najviše potražuje i ispisati na standardnom izlazu ukupnu količinu u kojoj se proizvodi. NAPOMENA: koristiti strukturu

```

typedef struct
{
    char ime[20];
    int kolicina;
} proizvod;

```

2. Napisati funkciju `void sifrat(char* rec, char* kljuc)` koja šifruje `rec` na sledeći način: za svako slovo reči `rec` i odgovarajuće slovo `kljuc` određuje koliki je (alfabetski) razmak između njih i označimo taj broj sa `k`. Potom to slovo `reci` zamenjuje `k`-tim slovom alfabeta. Podrazumeva se da je `kljuc` duži od `reci`. PRIMER: za `rec = bac` i `kljuc = dfge` rezultat je `bed`
3. Argument komadne linije je broj  $N$  ( $N \geq 2$ ). Napisati program koji na standardni izlaz ispisuje sledeću sliku: za  $N=2$ :

```

*
* *
* *

```

za  $N=3$ :

```

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

```

za  $N=4$ :

```

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

```

## Glava 6

# Akademska godina 2010/2011

### 6.1 Programiranje 1, kolokvijum, 27.11.2010.

1. Napisati URM program koji izračunava funkciju  $f(x, y) = x^y$ .
2. Napisati URM program i C program koji izračunavaju sledeću funkciju:

$$f(x, y) = \begin{cases} 2 * x & , x < y \\ x - y & , x \geq y \end{cases}$$

3. Sa standardnog ulaza unosi se četvorocifren broj N. Broj M se dobja od broja N zamenom cifri jedinica i hiljada i cifri desetica i stotina. Na standardni izlaz potrebno je ispisati **Jednaki su!** ukoliko su brojevi M i N jednaki, a u suprotnom **Nisu jednaki**.

Primer:

*ulaz:* 1331 *izlaz:* Jednaki su!

*ulaz:* 1231 *izlaz:* Nisu jednaki.

### 6.2 Programiranje 1, prvi mini-test na vežbama, 2010/2011

#### 6.2.1 Grupa 1

Napisati funkciju `int ind(int a[], int n)` koja kao povratnu vrednost ima indeks onog elementa niza koji je po vrednosti najbliži srednjoj vrednosti onih elemenata niza brojeva koji su deljivi sa 3.

Program testirati pozivom funkcije iz main programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza unosi broj n, a zatim niz od n celih brojeva (maksimalna dimenzija niza je 100 elemenata). Napr.

za *ulaz:* n = 5, a = 1, 2, 3, 4, 5 *ispis:* 2

za *ulaz:* n = 5, a = 3, 6, 2, 4, 7 *ispis:* 3

#### 6.2.2 Grupa 2

Napisati funkciju `void brojanje(int a[], int brojac[], int N)` čiji su argumenti `a` i `brojac` celobrojni nizovi dimenzije N. Vrednosti elemenata niza `a` su između 0 i N - 1. Funkcija izračunava elemente niza `brojac` tako da je `brojac[i]` jednak broju pojavljivanja broja `i` u nizu `a`. Program testirati pozivom funkcije iz main programa - korisnik učitava broj N i potom niz `a` dužine N, potom poziva funkciju i potom na standardnom izlazu izpisuje dobijeni niz.

### 6.2.3 Grupa 3

Implementirati funkciju `int min_max(int a[], int n)` koja prihvata celobrojni niz, pronalazi indekse najmanjeg i najvećeg elementa tog niza koristeći samo jedan prolaz (jednu petlju), a zatim kao povratnu vrednost vraća manji od ta dva indeksa.

Program testirati pozivom funkcije iz main programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza unosi niz dužine 10 elemenata.

### 6.2.4 Grupa 4

Napisati funkciju `int VratiPoziciju(int a[], int n, int b[], int m)` koja vraća prvu poziciju u nizu a na kojoj se pojavljuje broj iz niza b ili -1 ako a ne sadrži ni jedan broj iz b.

Napr.

Ako je  $a = \{3, -5, 7, 11\}$  i  $b = \{2, 4, 3, 11\}$  onda funkcija treba da vrati poziciju 0.

Ako je  $a = \{3, -5, 7, 11\}$  a  $b = \{11, -7, 2, 8\}$  onda funkcija treba da vrati poziciju 3.

Program testirati pozivom funkcije iz main programa i ispisom rezultata na standardni izlaz, pri čemu korisnik sa standardnog ulaza učitava broj  $n$  potom niz  $a$  dužine  $n$ , potom učitava broj  $m$  i potom niz  $b$  dužine  $m$ .

### 6.2.5 Grupa 5

Napisati funkciju `int novi_niz(int a[], int b[], int n, int c[])` koja na osnovu nizova  $a$  i  $b$  pravi niz  $c$  tako što se naizmenično raspoređuju elementi nizova  $a$  i  $b$ , tj.  $a_0, b_0, a_1, b_1, a_2, b_2, \dots$ . Nizovi  $a$  i  $b$  su iste dimenzije i njihova dimenzija je  $n$ . Funkcija vraća dimenziju niza  $c$ .

Funkciju testirati pozivom u main-u, pri čemu se sa standardnog ulaza učitava  $n$ , potom nizovi  $a$  i  $b$  dužine  $n$ . Nakon poziva funkcije na standardni izlaz istampati niz  $c$ .

## 6.3 Programiranje 1, drugi mini-test na vežbama, 2010/2011

### 6.3.1 Grupa 1

Napisati funkciju koja formira niz  $b[0], b[1], \dots$  od matrice tako što je svaki element niza jednak srednjoj vrednosti elemenata vrste matrice. Funkciju testirati pozivom u main-u, pri čemu korisnik sa standardnog ulaza učitava dimenzije matrice, a potom i elemente matrice. Dobijeni niz ispisati na standardni izlaz. Maksimalna dimenzija matrice je  $50 \times 50$ .

### 6.3.2 Grupa 2

Napisati funkciju u kojoj se razmenjuju elementi  $k$ -te i  $t$ -te vrste matrice ( $k$  i  $t$  su argumenti funkcije). Funkciju testirati pozivom u main-u i ispisom novodobijene matrice na standardni izlaz. Sa standardnog ulaza učitavaju se dimenzije matrice, a potom i elementi matrice i brojevi  $k$  i  $t$ . Maksimalna dimenzija matrice je  $50 \times 50$ .

### 6.3.3 Grupa 3

Napisati funkciju koja za datu kvadratnu matricu proverava da li je simetrična u odnosu na glavnu dijagonalu. Funkciju testirati pozivom u main-u, pri čemu korisnik sa standardnog ulaza učitava dimenziju matrice, potom kvadratnu matricu maksimalne dimenzije  $50 \times 50$ , a na standardni izlaz ispisuje poruku da li je matrica simetrična ili ne.

### 6.3.4 Grupa 4

Napisati funkciju koja za datu matricu  $A$  nalazi sve elemente matrice koji su jednaki zbiru svih svojih susednih elemenata i štampa ih u obliku (broj vrste, broj kolone, vrednost elementa). Funkciju testirati pozivom u main-u, pri čemu korisnik sa standardnog ulaza učitava dimenzije matrice, potom matricu maksimalne dimenzije 50x50.

### 6.3.5 Grupa 5

Napisati funkciju `void maksimalni(int mat[][50], int n, int* max)` kojom se u promenljivu `max` smešta najveći element u donjem trouglu matrice, uključujući i glavnu dijagonalu. Napisati funkciju kojom se sa standardnog ulaza učitava dimenzija kvadratne matrice, a potom i elementi matrice. Funkcije testirati pozivom u main-u, rezultat ispisati na standardni izlaz. Maksimalna dimenzija matrice je 50x50.

## 6.4 P1, završni ispit, januar 2011

1. Napisati URM program koji izračunava vrednost funkcije  $f(x) = 2^{(x+y)}$  (gde su  $x$  i  $y$  zadati kao argumenti u prva dva registra, a rezultat treba da bude smešten u prvi registar).
2. Napisati C funkciju koja u prosleđenom nizu eliminiše sve brojeve koji nisu deljivi svojim indeksom (vrednost na indeksu 0 zadržati, jer nije dozvoljeno deljenje sa 0). Niz reorganizovati, tako da nema *rupa* koje su nastale eliminacijom elemenata. Kao rezultat funkcije vratiti novu dimenziju niza. Na primer:  
niz  $a = \{4, 2, 1, 6, 7, 8, 10, 2, 16, 3\}$  se transformiše u niz  $a = \{4, 2, 6, 16\}$  i vraća se vrednost 4.
3. a) Napisati C funkciju `int procitaj_recenicu(char *s, int max_len)`, koja sa standardnog ulaza čita rečenicu i smešta je u nisku `s`. Čitanje rečenice se zaustavlja ako se pročita simbol `.` ili je već učitano `max_len-1` karaktera. Funkcija treba da vrati broj pročitanih karaktera.  
b) Napisati C funkciju `void prebroj(char *s, int *broj_malih, int *broj_velikih)`, koja za zadatau nisku `s` računa broj malih i velikih slova koji se u njoj pojavljuju.  
c) Napisati glavni program koji sa standardnog ulaza čita rečenice i na standardni izlaz ispisuje onu kod koje je razlika broja malih i velikih slova najveća.
4. a) Uvesti tip podataka `Tacka` kojim mogu da se predstavljaju tačke dekartovske ravni (zadate koordinatama  $x$  i  $y$  tipa `float`).  
b) Napisati funkciju `float rastojanje(Tacka a, Tacka b)` koja kao povratnu vrednost ima dužinu duži čije su krajnje tačke  $a$  i  $b$  (dužina duži između tačaka  $a(x_1, y_1)$  i  $b(x_2, y_2)$  može se računati po formuli  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ).  
c) Ime datoteke je argument komadne linije. Datoteka se sastoji iz dva reda, a u svakom redu su dva broja koja određuju jednu tačku. Napisati glavni program koji učitava tačke iz datoteke i u datoteku `duzina.txt` ispisuje dužinu duži određenu ovim tačkama.

## 6.5 P1, završni ispit, april 2011

1. Napisati urm program koji izračunava funkciju:

$$f(x, y) = \begin{cases} x/3 & , 3|x \\ y^2 & , inace \end{cases}$$

2. a) Uvesti tip podataka `Sifra` kojim se opisuje način šifrovanja alfanumeričkih karaktera. Svaka šifra se opisuje celobrojnomo vrednoscu `b` koja određuje broj pozicija pomeranja, kao i karakterom `'L'` ili `'D'` koji određuje smer pomeranja (levo ili desno).

- b) Napisati funkciju `void sifruj(char rec[], Sifra s)` koja transformiše zadatau reč `rec` po šifri `s`. Reč se šifruje tako što se svako slovo zamenjuje slovom za `b` mesta levo ili desno od njega u abecedi, i to ciklično, a isto tako i za cifre.  
Npr: za `b=2`, i `smer='D'` : a se menja sa c, b sa d,..., x sa z,y sa a, z sa b, 1 sa 3, .. 8 sa 0, 9 sa 1
- c) Načini šifrovanja se zadaju kao argumenti komandne linije i to u obliku `2 D 5 L`. Datoteka `ulaz.txt` se sastoji od alfanumerickih karaktera. Šifrovano prepisati sadržaj ove datoteke u datoteku `izlaz.txt`, redom po svim zadatim načinima šifrovanja.
3. a) Napisati funkciju  
`void svojstva(int a[][MAX], int m, int n, int *refl, int* sim, int* tranz)`  
 kojom se za zadatau relaciju (predstavljenu matricom `a`) ispitije da li je refleksivna, simetrična i tranzitivna. Funkcija postavlja na 1 odgovarajući indikator. Relacija se zadaje matricom `a` tako da važi: `i@j` akko `a[i][j]=1`
- b) Sa standardnog ulaza prvo se unose dimenzije matrice `m` i `n`, a nakon toga elementi matrice. Učitati matricu, i ispitati da li je relacija koju predstavlja relacija ekvivalencije (refleksivna, simetrična i tranzitivna).



# Glava 7

## Akademska godina 2011/2012

### 7.1 Programiranje 1 2011/2012, završni ispit, januar 2012

#### 7.1.1 Grupa A

##### Deo II (radi se na papiru)

1. Napisati URM program koji izračunava funkciju  $f(x, y, z) = \min(x, y, z)$ . (25 poena)

##### Deo II (radi se na računaru)

1. Napisati program koji učitava dve vrednosti tipa `double` i ispisuje njihov zbir na standardni izlaz. (5 poena; KVALIFIKACIONI ZADATAK — OSTALI ZADACI SE NE PREGLEDAJU UKOLIKO OVAJ NE RADI ISPRAVNO).
2. (30 poena) Implementirati funkciju `int strspn(char* s, char* t)` koja izračunava dužinu početnog dela niske `s` sastavljenog isključivo od karaktera sadržanih u niski `t`.  
Napisati i program koji sa standardnog ulaza učitava dve niske (dužine najviše 100 karaktera, svaku u zasebnom redu) i ispisuje rezultat poziva funkcije `strspn` na standardni izlaz.  
Na primer, za učitane podatke "734a.bf62", "0123456789") program ispisuje vrednost 3.

3. (40 poena) Napisati program koji učitava, a zatim i sabira (pozitivne) razlomke.
  - (a) Definisati tip podataka `RAZLOMAK` pogodan za predstavljanje (pozitivnih) razlomaka (čiji su imenilac i brojilac celi brojevi).
  - (b) Definisati funkciju `unsigned manji_ili_jednak(RAZLOMAK a, RAZLOMAK b)` koja vraća 1 ako je razlomak `a` manji ili jednak od razlomka `b`, a 0 inače.
  - (c) Definisati funkciju `RAZLOMAK saberi(RAZLOMAK a, RAZLOMAK b)` koja sabira dva razlomka i vraća rezultat.
  - (d) Definisati funkciju `unsigned ucitaj_razlomke(RAZLOMAK* razlomci, unsigned n)` koja učitava `n` puta po dva neoznačena cela broja (koje predstavljaju brojioce i imenioce razlomaka), na osnovu njih popunjava niz `razlomci` i vraća broj uspešno učitanih razlomaka.
  - (e) Definisati funkciju `RAZLOMAK maksimum(RAZLOMAK* razlomci, unsigned n)` koja izračunava maksimalni razlomak iz niza `n` razlomaka iz datog niza i vraća rezultat.
  - (f) Definisati funkciju `RAZLOMAK saberi_razlomke(RAZLOMAK* razlomci, unsigned n)` koja sabira `n` razlomaka iz datog niza razlomaka i vraća rezultat.

- (g) Definisati funkciju `main` u kojoj se sa standardnog ulaza učitava celobrojna nenegativna vrednost `N` i dinamički se alokira prostor za `N` objekata tipa `RAZLOMAK`.

Ukoliko nije uspela dinamička alokacija, funkcija završava sa radom.

Inače, poziva se funkcija `ucitaj_razlomke` koja u alocirani prostor smešta razlomke sa standardnog ulaza. Ukoliko je uspešno učitano `m` razlomaka (`N` ne mora da bude jednako `m`), onda se poziva funkcija `maksimum` za `m` učitanih razlomaka i ispisuje njen rezultat u obliku `brojilac/imenilac` na standardni izlaz (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol `?`). Posle toga se poziva funkcija `saberi_razlomke` za `m` učitanih razlomaka i ispisuje njen rezultat u obliku `brojilac/imenilac` (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol `?`).

## 7.1.2 Grupa B

### Deo II (radi se na papiru)

1. Napisati URM program koji izracunava funkciju  $f(x, y, z) = x + y + z$  (25 poena)

### Deo II (radi se na računaru)

1. Napisati program koji ucitava dve vrednosti tipa `double` i ispisuje na standardni izlaz vrednost "1" ako je prva strogo veće, a "2" inače. (5 poena; KVALIFIKACIONI ZADATAK — OSTALI ZADACI SE NE PREGLEDAJU UKOLIKO OVAJ NE RADI ISPRAVNO).

2. (25 poena) Implementirati funkciju `int strcspn(char* s, char* t)` koja izracunava poziciju prvog karaktera niske `s` koji je sadržan u niski `t`.

Napisati i program koji sa standardnog ulaza učitava dve niske (dužine najviše 100 karaktera, svaku u zasebnom redu) i ispisuje rezultat poziva funkcije `strcspn` na standardni izlaz.

Na primer, za učitane podatke `strcspn("ab.c73#b62", "0123456789")` program ispisuje vrednost 4.

3.
  - Definisati tip podataka `TACKA` pogodan za predstavljanje tačke Dekartovske ravni (čije su `x` i `y` koordinate podaci tipa `double`).
  - Definisati funkciju `double rastojanje(TACKA a, TACKA b)` koja izracunava rastojanje izmedju dve tačke.
  - Definisati funkciju `unsigned ucitaj_poligon(TACKA* tacke, unsigned n)` koja učitava `n` puta po dve vrednosti tipa `double` (koje predstavljaju koordinate temena poligona) i upisuje ih u zadati niz tačaka. Funkcija vraća broj uspešno učitanih tačaka.
  - Definisati funkciju `double obim(TACKA* poligon, unsigned n)` koja izracunava obim poligona sa `n` tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
  - Definisati funkciju `double maksimalna_stranica(TACKA* poligon, unsigned n)` koja izracunava dužinu najduže stranice poligona sa `n` tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
  - Definisati funkciju `main` u kojoj se sa standardnog ulaza učitava celobrojna nenegativna vrednost `N` i dinamički se alokira prostor za `N` objekata tipa `TACKA`.

Ukoliko nije uspela dinamička alokacija, funkcija završava sa radom.

Inače, poziva se funkcija `ucitaj_poligon` koja u alocirani prostor smešta tačke sa standardnog ulaza. Ukoliko je uspešno učitano `m` tačka (`N` ne mora da bude jednako `m`), onda se poziva funkcija `obim` za `m` učitanih tačaka i ispisuje njen rezultat na standardni izlaz (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol `?`). Posle toga se poziva funkcija `maksimalna_stranica` za `m` učitanih tačaka i ispisuje njen rezultat na standardni izlaz (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol `?`).

## 7.2 Programiranje 1 2011/2012, završni ispit, februar 2012

### 7.2.1 Deo II (radi se na papiru)

1. Napisati URM program koji izracunava funkciju

$$f(x, y, z) = \begin{cases} 1, & \text{ako je } x + y > z \\ 2, & \text{inače} \end{cases}$$

### 7.2.2 Deo II (radi se na računaru)

1. Napisati program koji za unete brojeve a11, a12, a21, a22 tipa `float` izračunava i ispisuje na standardni izlaz determinantu matrice:

```
a11 a12
a21 a22
```

Pri ispisu vrednosti se zaokružuju na 4 decimale.

Primer 1:

```
Ulaz: 1 2 3 4
Izlaz: -2.0000
```

Primer 2:

```
Ulaz: -1 0 0 1
Izlaz: -1.0000
```

Primer 3:

```
Ulaz: 1.5 -2 3 4.5
Izlaz: 12.7500
```

Primer 4:

```
Ulaz: 0.01 0.01 0.5 7
Izlaz: 0.0650
```

2. Napisati program koji ispisuje najveći od uneta 3 cela broja.

Primer:

```
Ulaz: -1000 256 0
Izlaz: 256
```

3. Napisati program koji ispisuje vrednost funkcije  $\cos(x)$  u 10 ravnomerno razmaknutih tačaka intervala  $[a, b]$  ( $a$  i  $b$  su vrednosti tipa `double`, za koje važi  $a < b$  i učitavaju se sa tastature). Pri ispisu vrednosti se zaokružuju na 4 decimale. Za neispravan unos, program ispisuje broj -1.

Primer 1:

```
Ulaz: 1 10
```

```
Izlaz: 0.5403 -0.4161 -0.9900 -0.6536 0.2837 0.9602 0.7539 -0.1455 -0.9111 -0.0000
```

```
Ulaz: 0 28.274
```

```
Izlaz: 1.0000 -1.0000 1.0000 -1.0000 1.0000 -1.0000 1.0000 -1.0000 1.0000 -1.0000
```

```
Ulaz: 1 -3
```

```
Izlaz: -1
```

```
Ulaz: 0 1
```

```
Izlaz: 1.0000 0.9938 0.9754 0.9450 0.9028 0.8496 0.7859 0.7125 0.6303 0.5403
```

4. Napisati program koji za unete vrednosti sat, minut i sekund (tipa `unsigned int`) izračunava koliko je sekundi preostalo do narednog podneva i rezultat ispisuje na standardni izlaz. Za neispravan unos, program ispisuje broj -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz: 9 0 0	Ulaz: 12 0 1	Ulaz: 17 24 6	Ulaz: 12 7 87
Izlaz: 10800	Izlaz: 86399	Izlaz: 66954	Izlaz: -1

5. Sa standardnog ulaza unosi se broj  $n$ . Napisati program koji ispisuje brojeve od 1 do  $n$ , zatim od 2 do  $n - 1$ , 3 do  $n - 2$ , itd. Za neispravan unos, program ispisuje broj -1.

Primer 1:  
 Ulaz: 5  
 Izlaz: 1 2 3 4 5 2 3 4 3

Primer 2:  
 Ulaz: -4  
 Izlaz: -1

Primer 2:  
 Ulaz: 7  
 Izlaz: 1 2 3 4 5 6 7 2 3 4 5 6 3 4 5 4

Primer 4:  
 Ulaz: 3  
 Izlaz: 1 2 3 2

6. Implementirati funkciju `int str_str(char s[], char t[])` koja proverava da li niska `s` sadrzi nisku `t`. Zatim napisati program koji sa standardnog ulaza učitava pet redova (svaki red ima najviše 100 karaktera) i koji ispisuje sve redne brojeve linija koje sadrže nisku `program` (linije se numerišu od broja 1). Ukoliko ne postoji red sa niskom `program` ispisati -1.

Primer 1:  
 Ulaz:  
 novi red\*nprogram  
 c prog. jezik  
 c? programskih jezik  
 Programski odbor  
 <b>program</b>

Izlaz: 1 3 5

Primer 2:  
 Ulaz:  
 PROGRAMIRANJE  
 program1.c  
 grupa2  
 Programator  
 Paralelno programiranje

Izlaz: 2 5

Primer 3:  
 Ulaz:  
 Programiranje  
 algebra  
 linearno prograMiranje  
 prvi\_pro\_gram.c  
 progres americkih trupa

Izlaz: -1

7. (a) Definisati strukturu `Tacka` za predstavljanje tačaka u ravni sa koordinatama tipa `double`.  
 (b) Napisati funkciju `Rastojanje` koja izračunava rastojanje između dve tačke.  
 (c) Napisati funkciju `Obim` koja izračunava obim poligona zadatog nizom tačaka.  
 (d) Napisati funkciju `Povrsina` koja izračunava površinu poligona zadatog nizom tačaka (koristiti Heronov obrazac:  $P = \sqrt{s(s-a)(s-b)(s-c)}$  pri čemu je `s` poluobim trougla sa stranicama `a`, `b` i `c`).  
 (e) Sa ulaza se unosi broj `n` i zatim `n` tačaka koje redom predstavljaju koordinate temena konveksnog prostog poligona. Koristiti dinamičku alokaciju memorije. Program poziva funkcije `Obim` i `Povrsina` i ispisuje na standardni izlaz obim i površinu poligona. Za neispravan unos ili neuspešnu alokaciju memorije program ispisuje broj -1. Pri ispisu vrednosti se zaokružuju na 4 decimale.

Primer 1:  
 Uneti broj tacaka:  
 4  
 Uneti koordinate tacaka:  
 0 0  
 0 4  
 6 4  
 6 0

Obim: 20.0000  
 Povrsina: 24.0000

Primer 3:  
 Uneti broj tacaka:  
 6  
 Uneti koordinate tacaka:  
 -1 1  
 0 2  
 1 1  
 1 -1  
 0 -2  
 -1 -1

Obim: 9.6569  
 Povrsina: 6.0000

Primer 2:  
 Uneti broj tacaka:  
 4  
 Uneti koordinate tacaka:  
 -1 1  
 2 3  
 4 1  
 0 -7

Obim: 23.4405  
 Povrsina: 25.0000

Primer 4:  
 Uneti broj tacaka:  
 -3  
 Izlaz: -1

## 7.3 Programiranje 1 2011/2012, završni ispit, septembar 2012

### 7.3.1 Deo II (radi se na papiru)

1. Napisati URM program koji izracunava broj 1331 i smešta ga u prvi registar.

### 7.3.2 Deo II (radi se na računaru)

1. Napisati program koji za unete koeficijente  $A$  i  $B$  (tipa `int`) izračunava rešenje jednačine  $Ax + B = 0$ . Program treba da prijavi ukoliko jednačina nema rešenja ili ima više od jednog rešenja.
2. Napisati program koji za uneti označeni ceo broj  $a$  i neoznačeni ceo broj  $b$  skraćuje razlomak  $\frac{a}{b}$  i ispisuje brojilac i imenilac skraćenog razlomka.
3. Napisati program koji za uneto  $n$  ispisuje „trougao” sačinjen od „koordinata” svojih tačaka. Rezultat rada za  $n = 1, 2, 3, 4$  je:

$n = 1$	$n = 2$	$n = 3$	$n = 4$
(1,1)	(1,2) (2,2)	(1,3) (2,3) (3,3)	(1,4) (2,4) (3,4) (4,4)
	(1,1)	(1,2) (2,2)	(1,3) (2,3) (3,3)
		(1,1)	(1,2) (2,2)
			(1,1)

4. Napisati program koji korišćenjem aritmetičkih operacija razmenjuje prvu i poslednju cifru unetog neoznačenog celog broja (na primer, za unos 1234 program ispisuje 4231).

- Implementirati funkciju `char* my_strpbrk(char* str1, char* str2)` koja vraća pokazivač na prvo pojavljivanje u okviru niske `str1` nekog karaktera niske `str2`, odnosno `NULL` pokazivač ukoliko `str1` ne sadrži ni jedan od karaktera iz `str2`. Napisati zatim i program koji korišćenjem date funkcije ispisuje sva pojavljivanja samoglasnika u rečenici (dužine najviše 100 karaktera) koja se učitava sa standardnog ulaza.
- Definisati strukturu `student` koja opisuje ime i prezime studenta kao i njegov broj poena na ispitu. Napisati funkciju koja za datog studenta, na osnovu broja poena automatski određuje zaključnu ocenu, na osnovu trenutno važećih pravila na fakultetu. Napisati funkciju koja u nizu studenata određuje ukupan broj studenata koji su položili ispit. Napisati program koji omogućava unos podataka o studentima (za svakog studenta se unos i ime, prezime i broj poena, odvojeni proizvoljnim brojem belina, a unos se završava kada se unese prazan red), nakon čega se za svakog studenta ispisuje ime prezime i ocena, i ispisuje se ukupan broj studenata koji su položili ispit kao i ukupan broj studenata koji su pali ispit.

## 7.4 Programiranje 1 2011/2012, završni ispit, oktobar 2012

- Napisati URM program koji izračunava funkciju  $f(x, y) = \lfloor \frac{2x}{3} \rfloor$
- Napisati program koji za unetu vrednost poluprečnika kruga ispisuje njegov obim i površinu. Proveriti da li je uneta vrednost korektna. Rezultat prikazati na 5 decimala. Koristiti konstante definisane u standardnoj biblioteci.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
ulaz: 4.5	ulaz: 3.58	ulaz: 11.23	ulaz: 23.14
izlaz: 28.27433 63.61725	izlaz: 22.49380 40.26391	izlaz: 70.56017 396.19533	izlaz: 145.39290 1682.19586

- Sa standardnog ulaza unosi se ceo pozitivan broj `n`. Pretpostavlja se da je unos ispravan. Napisati program koji stampa zvezdice i minuseve sledeceg oblika za:

n = 4	n = 3	n = 2	n = 6
***** ***----- **-----* *-----* **-----* ***----- *****	***** **----- *-----* **-----* *****	**** *--* ****	***** ***** ***** ***-----* **-----* *-----* **-----* ***-----* ***** ***** *****

- Sa standardnog ulaza se unosi ceo neoznačeni broj `n`, a zatim i niz od `n` elemenata celobrojnog tipa (njih najviše 100). Napisati program koja kvadrira one elemente niza koji su veći od aritmetičke sredine niza, a zatim štampa modifikovani niz. U slučaju neregularnog unosa stampa se poruka: `GRESKA`.

Primer 1:	Primer 2:
ulaz: 5 1 6 7 2 3	ulaz: 7 23 4 67 -34 56 12 89

izlaz: 1 36 49 2 3

izlaz: 23 4 4489 -34 3136 12 7921

Primer 3:

Primer 4:

ulaz: 4 12 789 134 390

ulaz: 102

izlaz: 12 622521 134 152100

izlaz: GRESKA

5. Napisati funkciju `unsigned btoi(char* s, unsigned char b)` koja određuje vrednost zapisa datog neoznačenog broja `s` u datoj osnovi `b`. Napisati funkciju `void itob(unsigned n, unsigned char b, char* s)` koja datu vrednost `n` zapisuje u datoj osnovi `b` i smešta rezultat u nisku `s`. Napisati zatim program koji čita liniju po liniju sa standardnog ulaza i obrađuje ih sve dok ne naiđe na praznu liniju. Svaka linija sadrži jedan dekadni, oktalni ili heksadekadni broj (zapisan kako se zapisuju konstante u programskom jeziku C). Program za svaki uneti broj ispisuje njegov binarni zapis. Pretpostaviti da će svi uneti brojevi biti u opsegu tipa `unsigned`.

Primer 1:

Primer 2:

Primer 3:

Primer 4:

0x49

012

123

981

1001001

1010

1111011

1111010101

0x1ABC

435

0777

1101010111100

110110011

111111111

0x64FE

110010011111110

6. Napisati program koji izracunava razliku dva broja koji nastaju rotiranjem u levo i rotiranjem u desno cifara unetog celog pozitivnog broja. (Pretpostaviti da je unos ispravan.)

Primer 1:

Primer 2:

Primer 3:

Primer 4:

ulaz: 1234

ulaz: 238

ulaz: 352112

ulaz: 78

izlaz: -1782

izlaz: -441

izlaz: 285912

izlaz: 0

objasnjenje: 2341 - 4123

7. Definisati strukturu `PROIZVOD` koja opisuje proizvode u magacinu. Svaki proizvod se opisuje svojim imenom, cenom (tipa `float`) i kolicinom (tipa `int`).

- Napisati funkciju koja određuje srednju cenu proizvoda u magacinu (ne uzimajući u obzir njihove kolicine)
- Napisati funkciju koja određuje proizvode čija je ukupna vrednost (cena\*kolicina) najveća odnosno najmanja.
- Napisati glavni program gde se sa standardnog ulaza najpre unosi broj proizvoda (broj proizvoda nije unapred poznat, niti ograničen), a zatim se unose podaci o proizvodima. Koriscenjem funkcija izracunati, a zatim na standardni izlaz ispisati srednju cenu proizvoda, kao i ime i ukupnu vrednost onih proizvoda čija je ukupna vrednost najveća/najmanja

Primer 1:

Primer 2:

Primer 3:

Primer 4:

ulaz:

ulaz:

ulaz:

ulaz:

1

2

3

4

secer 100.50 1000

ulje 183.40 200

kafa 85 200

mleko 81 1000

izlaz:	cips 83 450	hleb 50 450	pivo 90 700
100.5	izlaz:	plazma 100 300	brasno 70 300
secer 100500	133.2	izlaz:	jogurt 100 200
secer 100500	cips 37350	78.33	izlaz:
	ulje 36680	plazma 30000	85.25
		kafa 17000	mleko 81000
			jogurt 20000



## Glava 8

# Akademska godina 2012/2013

### 8.1 Test 4 - zadaci, 02.12.2012.

#### 8.1.1 Grupa 1

(3 poena) 1. Sa standardnog ulaza unose se broj  $n$  i  $n$  celih brojeva. Odrediti maksimum unetih brojeva i rezultat ispisati na standardni izlaz.

```
ulaz: 6 1 34 -21 89 2 0      ulaz: 4 67 1 34 -90      ulaz: 1 8      ulaz: 0
izlaz: 89                    izlaz: 67                izlaz: 8      izlaz:
```

(7 poena) 2. Napisati program koji sa ulaza učitava pozitivne cele brojeve sve dok se ne unese 0. Svaki uneti broj se deli na proste činioce (ne računajući broj 1, npr.  $14 = 2 * 7$ ). Nakon unosa, na standardni izlaz se ispisuje zbir svih prostih činilaca.

```
ulaz: 6 19 21 0              ulaz: 0                ulaz: 100 14 23 76 18 0      ulaz: 34 25 15 0
izlaz: 34                    izlaz: 0                izlaz: 77                    izlaz: 37
```

#### 8.1.2 Grupa 2

(3 poena) 1. Sa standardnog ulaza unosi se  $n$  i  $n$  celih brojeva. Na standardni izlaz ispisati sumu unetih brojeva.

```
ulaz: 6 1 34 -21 89 2 0      ulaz: 4 67 1 34 -90      ulaz: 1 8      ulaz: 0
izlaz: 105                    izlaz: 16                izlaz: 8      izlaz: 0
```

2. (7 poena) Napisati program koji uneti neoznačeni ceo broj transformiše tako što svaku parnu cifru u zapisu broja uveća za 1.

```
ulaz: 22                      ulaz:1230              ulaz: 0                ulaz: 23456
izlaz: 33                     izlaz: 1331            izlaz: 1                izlaz: 33557
```

#### 8.1.3 Grupa 3

1. (3 poena) Sa standardnog ulaza unosi se  $n$  i  $n$  brojeva. Na standardni izlaz ispisati broj pozitivnih brojeva.

ulaz: 6 1 34 -21 89 2 0

ulaz: 4 67 1 34 -90

ulaz: 1 8

ulaz: 0

izlaz: 6

izlaz: 4

izlaz: 0

izlaz: 0

2.

(7 poena)

Napisati program koji za uneti neoznačeni ceo broj N, ispisuje zvezdice na sledeći način:

N = 5

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

N = 4

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

N = 3

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

N = 0

## Programiranje 1 2011/2012, Test 5, decembar 2012

### Grupa 1

(1 poen)

**Zadatak 1** Napisati funkciju `int suma(int a[], int n)` koja izračunava sumu svih elemenata niza `a`, dužine `n`. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(2 poena)

**Zadatak 2** Napisati funkciju `int mala_velika(char s[], int* m, int* v)` koja određuje broj malih i broj velikih slova u nisku. Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

(3 poena)

**Zadatak 3** Napisati funkciju `int f3(int a[], int n, int b[], int m)` i ispituje da li prvi sadrži bar dva broja koji se pojavljuju u drugom nizu. Povratna vrednost je dakle, 0, ili 1. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(4 poena)

**Zadatak 4** Napisati funkciju `void ukloniSlova(char s[])` koja iz niske `s` uklanja sva mala i velika slova. Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

### Grupa 2

(1 poen)

**Zadatak 5** Napisati funkciju `int f1(char s[])` koja prihvata tu nisku i proverava da li niska sadrži veliko slovo. Funkcija vraća 1 ako sadrži veliko slovo, inače 0. Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

(2 poena)

**Zadatak 6** Napisati funkciju `void minimax(int a[], int n, int* min, int *max)` koja pronalazi najmanji i najveći element niza. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(3 poena)

**Zadatak 7** Napisati funkciju `int pp(char s[], char t[])` koja određuje poziciju poslednjeg karaktera niske `s` sadržanog u okviru niske `t`, zanemarujući pri tom razliku između velikih i malih slova, ili `-1` ako takvog karaktera nema. Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

```
pp("a4BA3Bc", "A3b") = 5
```

(4 poena)

**Zadatak 8** Napisati funkciju `int f4(int a[], int n, int b[], int m)` koja modifikuje prvi niz tako da ne sadrži ni jedan element drugog niza. Funkcija vraća novu dimenziju niza. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

### Grupa 3

(1 poen)

**Zadatak 9** Napisati funkciju `void ispis(int niz[], int n)` koja ispisuje svaki drugi element datog niza počevši od nazad. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(2 poena)

**Zadatak 10** Napisati funkciju `void prosek(int niz[], int n, float* pr, int* br)` koja računa prosek elemenata datog niza i upisuje u promenljivu `pr` i racuna broj elemenata vecih od proseka i upisuje u promenljivu `br`. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(3 poena)

**Zadatak 11** Napisati funkciju `verb unija(int a[], int n, int b[], int m, int c[], int* k)` koja u `c` upisuje uniju skupova `a` i `b`, a u `k` dobijenu dimenziju niza `c`. Skupovi su sortirani, te i rezultujući niz treba da bude sortiran. Testirati pozivom u `main-u`. Maksimalna dužina niza je 100 elemenata.

(4 poena)

**Zadatak 12** Napisati funkciju `void min_razlika(char s[], char s1[], char s2[])` koja u datotoj nisci `s` pronalazi dve reči koje imaju minimalnu razliku između svojih samoglasnika. ( Reč je niz karaktera između dve praznine; razmak između samoglasnika reči `dan`as i `jutro` je 2, a razmak između `sutrk` i `mnozenje` je 5). Testirati pozivom u `main-u`. Maksimalna dužina niske je 20 karaktera.

## 8.2 I smer, Programiranje 1 2012/2013, završni ispit, ferbruar 2013

(3 poena)

**Zadatak 13** Napisati program koji izračunava i ispisuje zbir četiri najveća od pet brojeva unetih sa standardnog ulaza.

Primer 1: 0 2 3 4 5	Primer 2: 0 -5 100 23 3	Primer 3: -300 -2000 -19 -2 -1	Primer 4: 23 467 908 0 -1
14	126	-322	1398

(3 poena)

**Zadatak 14** Napisati program koji određuje i ispisuje zbir prve i poslednje cifre unetog neoznačenog celog broja. Ako je broj jednocifren, ispisati -1.

Primer 1: 12345	Primer 2: 3	Primer 3: 99	Primer 4: -56
6	-1	18	4

(3 poena)

**Zadatak 15** Napisati program koji za argument komandne linije  $n$  ispisuje sve brojeve od 1 do  $n$ , zatim svaki drugi broj od 1 do  $n$ , zatim svaki treći broj od 1 do  $n$  itd., završavajući sa svakim  $n$ -tim (tj. samo sa 1). U slučaju greške ispisati -1.

Primer 1: za n=3	Primer 2: za n=1	Primer 3: za n=7	Primer 4: za n=-23
1 2 3 1 3 1	1	1 2 3 4 5 6 7 1 3 5 7 1 4 7 1 5 1 6 1 7 1	

(3 poena)

**Zadatak 16** Sa standardnog ulaza se unosi jedna linija teksta. Napisati program koji prikazuje koliko puta se javilo svako od slova engleske abecede (praviti razliku između velikih i malih slova).

Primer 1:  
MaLoVeLiKo

```
a:1 b:0 c:0 d:0 e:1 f:0 g:0 h:0 i:1 j:0 k:0 l:0 m:0 n:0 o:2 p:0 q:0 r:0 s:0 t:0 u:0 v:0 w:0
x:0 y:0 z:0 A:0 B:0 C:0 D:0 E:0 F:0 G:0 H:0 I:0 J:0 K:1 L:2 M:1 N:0 O:0 P:0 Q:0 R:0 S:0 T:0
U:0 V:1 W:0 X:0 Y:0 Z:0
```

---

Primer 2:  
PrOGraMiRanje 1 1sPit.\*\*\*A

```
a:2 b:0 c:0 d:0 e:1 f:0 g:0 h:0 i:2 j:1 k:0 l:0 m:0 n:1 o:0 p:0 q:0 r:2 s:1 t:1 u:0 v:0 w:0
x:0 y:0 z:0 A:1 B:0 C:0 D:0 E:0 F:0 G:1 H:0 I:0 J:0 K:0 L:0 M:1 N:0 O:0 P:2 Q:0 R:1 S:0 T:0
U:0 V:0 W:0 X:0 Y:0 Z:0
```

---

Primer 3:  
Mzzz Abbb 3 yyy

```
a:0 b:3 c:0 d:0 e:0 f:0 g:0 h:0 i:0 j:0 k:0 l:0 m:0 n:0 o:0 p:0 q:0 r:0 s:0 t:0 u:0 v:0 w:0
x:0 y:3 z:3 A:1 B:0 C:0 D:0 E:0 F:0 G:0 H:0 I:0 J:0 K:0 L:0 M:1 N:0 O:0 P:0 Q:0 R:0 S:0 T:0
U:0 V:0 W:0 X:0 Y:0 Z:0
```

---

Primer 4:  
MalemAlIcK1 FaKulteT.

a:2 b:0 c:0 d:0 e:2 f:0 g:0 h:0 i:1 j:0 k:1 l:0 m:1 n:0 o:0 p:0 q:0 r:0 s:0 t:1 u:1 v:0 w:0  
x:0 y:0 z:0 A:1 B:0 C:1 D:0 E:0 F:1 G:0 H:0 I:0 J:0 K:1 L:0 M:1 N:0 O:0 P:0 Q:0 R:0 S:0 T:1  
U:0 V:0 W:0 X:0 Y:0 Z:0

---

(6 poena)

**Zadatak 17** Napisati funkciju `void ukloni(char *s)`; koja iz niske uklanja sva slova iza kojih neposredno sledi slovo koje je u abecedi nakon njih (veličina slova se zanemaruje). Testirati funkciju u programu koji učitava liniju teksta (najviše 100 karaktera).

Primer 1: zdRaVo svIma	Primer 2: 12345Abcd	Primer 3: JeD1aN D52Va.	Primer 4: abcd efg
zRVo vma	12345D	JeD1N D52Va.	d g

(7 poena)

**Zadatak 18** U datoteci čije se ime navodi kao argument komandne linije programa nalazi se broj *n*, a zatim *i* n reči (dužine najviše 50 karaktera). Napisati program koji učitava ovaj niz *i*

1. ispisuje ga, [3]
2. u datoteku `rez.txt` upisuje sve reči koje sadrže prvu reč *i* podvlaku. [4]

U slučaju greške ispisati -1.

Primer 1: ./a.out dat1.txt dat1.txt: 7 rec Opet _rec Reci rec_enica DVa recica_  rec Opet _rec Reci rec_enica DVA recica_  rez.txt: _rec rec_enica recica_		Primer 2: .\a.out dat2.txt dat2.txt: 11 Sunce sija iznad grada Sunce_Moje Jedan Dva Su_nce Sve Sunce123_123 s  Sunce sija iznad grada Sunce_Moje Jedan Dva Su_nce Sve Sunce123_123 s  rez.txt: Sunce_Moje Sunce123_123
--	--	--

---

Primer 3: ./a.out dat3.txt dat.txt: 18 Na danasnji dan roden je poznati engleski pisac Carls Dickens, a umro reformator srpskog jezika Vuk Stefanovic Karadzic.  Na danasnji dan roden je poznati engleski pisac Carls Dickens, a umro reformator srpskog jezika Vuk Stefanovic Karadzic.  rez.txt:		Primer 4: .\a.out  -1
---	--	--------------------------------

## 8.3 I smer, Programiranje 1 2012/2013, završni ispit, septembar 2013

1. Sa standardnog ulaza se unosi broj  $n$  i zatim  $n$  brojeva tipa `int`. Napisati program koji izračunava i ispisuje raspon (pozitivnu razliku između najmanjeg i najvećeg od njih). U slučaju greške ispisati -1.

Primer 1: 5 -3 4 -5 8 2  13	Primer 2: -2  -1	Primer 3: 7 -12 6 7 -2 12 0 15  27	Primer 4: 3 23 50 31  27
--------------------------------------	---------------------------	---	-----------------------------------

2.  $A_0$  papir ima površinu  $1m^2$  i odnos stranica  $1 : \sqrt{2}$ .  $A_1$  papir dobija se podelom papira  $A_0$  po dužoj ivici.  $A_2$  papir dobija se podelom  $A_1$  papira po dužoj ivici itd. Napisati program koji za uneto  $k$  ispisuje dimenzije papira  $A_k$  u milimetrima.

Primer 1: 4  297 210	Primer 2: 3  297 420	Primer 3: 7  74 105	Primer 4: 9  37 52
-------------------------------	-------------------------------	------------------------------	-----------------------------

3. Napisati funkciju `unsigned htoi(char s[])`; koja izračunava vrednost heksadekadnog broja zadatog u nizu karaktera `s` (pretpostavlja se da se vrednost može predstaviti tipom `unsigned`). Napisati program koji testira ovu funkciju učitavajući nisku `s` sa standardnog ulaza (ne više od 8 karaktera). Program treba da proveriti i da li je niska `s` ispravan heksadekadni zapis broja (ne praviti razliku između velikih i malih slova) i u slučaju greške da ispiše -1.

Primer 1: 3fA8  16296	Primer 2: 6D7F  28031	Primer 3: 8a06b  565355	Primer 4: 07f  -1
--------------------------------	--------------------------------	----------------------------------	----------------------------

4. Šifrovanje teksta vrši se tako što se karakter `a` menja karakterom `Z`, `b` karakterom `Y`, `c` karakterom `X`, ..., `y` karakterom `B`, i `z` karakterom `A`. Slično, `A` se menja karakterom `z`, ..., `Z` karakterom `a`. Svi karakteri osim cifara i slova se brišu. Napisati funkciju `void sifruj(char* s)`; koja vrši šifrovanje date niske. Napisati i program koji učitava tekst sa standardnog ulaza liniju po liniju (najviše 100 karaktera), vrši šifrovanje svake linije i rezultat ispisuje na standardni izlaz. Npr.

Primer 1: Programiranje 1  kILTIZNRIZMQV1  ----- Primer 3: Ispit, 5.9.2013.  rHkRG592013	Primer 2: Analiza..  zMZORAZ  ----- Primer 4: _Eksperiment_  vPHkVIRNVMG
---	---

5. Iz datoteke čije se ime navodi kao prvi argument komandne linije programa učitava se broj  $n$ , a zatim i kvadratna matrica koja sadrži brojeve tipa `double` dimenzije  $n \times n$ . Napisati program koji izračunava i u datoteku `rezultat.txt` upisuje razliku (na dve decimale) između zbira elemenata gornjeg trougla i zbira elemenata donjeg trougla matrice – gornji trougao čine svi elementi iznad sporedne dijagonale (ne računajući dijagonalu), a donji trougao čine svi elementi ispod sporedne dijagonale (računajući dijagonalu). U slučaju greške u datoteku upisati `GRESKA`.

Primer 1:  
./a.out ulaz.txt  
ulaz.txt:  
3  
2 3.2 4  
7 8.8 1  
2.3 1 1  
  
rezultat.txt:  
-2.10

Primer 2:  
./a.out matrica.txt  
matrica.txt:  
4  
2.3 1 12 8  
4 -8.2 7 14.5  
1 -2.5 9 11  
3 4.3 -5.7 2  
  
rezultat.txt:  
49.4

Primer 3:  
./a.out  
  
rezultat.txt:  
GRESKA

Primer 4:  
./a.out dat.txt  
dat.txt:  
-4  
  
rezultat.txt:  
GRESKA

## Glava 9

# Akademski godina 2013/2014

### 9.1 Test 2, 07.11.2013

#### 9.1.1 Grupa I

**Zadatak 19** Napisati URM program koji izračunava funkciju  $f(x, y, z) = 2 \cdot (x + y + z)$

**Zadatak 20** Sa standardnog ulaza se unosi četvorocifren ceo broj. Napisati program koji za uneti broj ispisuje na izlazu njegovu najveću cifru. Ako broj nije četvorocifren ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3842	-4239	123	-45678
8	9	-1	-1

**Zadatak 21** Sa standardnog ulaza se unosi 5 karaktera. Ispisati na izlazu broj unetih malih slova.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
aBcDe	Ba4b_	abcde	B6(AV
3	2	5	0

#### 9.1.2 Grupa II

**Zadatak 22** Napisati URM program koji izračunava funkciju:  $f(x, y, z) = \max(x, y, z)$ .

**Zadatak 23** Sa standardnog ulaza se unosi četvorocifren ceo broj. Napisati program koji datom broju razmenjuje najmanju i najveću cifru. Dobijeni broj ispisati na izlaz. Ako broj nije četvorocifren ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3842	-4239	123	-45678
3248	-4932	-1	-1

**Zadatak 24** Sa standardnog ulaza se unosi 5 karaktera. Ispisati na izlazu koliko se puta pojavilo veliko ili malo slovo a.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
aBcAe	aa4A_	aAaAa	B6(vV
2	3	5	0



### 9.1.3 Grupa III

**Zadatak 25** Napisati kod za URM program koji izračunava funkciju:

$$f(x, y) = \begin{cases} 2 \cdot x + 1 & , x \leq y \\ x - y & , \text{inace} \end{cases}$$

**Zadatak 26** Sa standardnog ulaza se unose 5 karaktera. Ispisati na izlazu koliko puta su se pojavile cifre.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
A1cA3	2a45_	43986	Ba(vV
2	3	5	0

**Zadatak 27** Sa standardnog ulaza se unose tri neoznačena trocifrena broja. Spojiti dva najveća u šestocifren broj. Spajanje izvršiti tako da najveći od trocifrenih brojeva bude na početku šestocifrenog broja. Dobijeni šestocifreni broj ispisati na izlazu. Ako neki od unetih brojeva nije trocifren, ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
384	123	1242	374
123	345	234	23
245	5	324	898
384245	-1	-1	-1

## 9.2 Test 4, 23.11.2013

### 9.2.1 Grupa I

**Zadatak 28** Sa standardnog ulaza unosi se ceo pozitivan broj  $N$ , a potom  $N$  celih brojeva. Odrediti proizvod onih brojeva koji su neparni i negativni. U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	-3	10	6
-2 -3 5 1 -1		-3 6 11 -20 -25 -8 42 -37 1 6	1 -19 -2 7 -27 -3
3	-1	-2775	-1539

**Zadatak 29** Sa standardnog ulaza unosi se ceo pozitivan neparan broj  $n$ . Napisati program koji za uneto  $n$  izračunava:

$$S = 1 \cdot 3 \cdot 5 - 1 \cdot 3 \cdot 5 \cdot 7 + 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 - 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot 11 + \dots (-1)^{\frac{n-1}{2}+1} \cdot 1 \cdot 3 \cdot \dots \cdot n$$

U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
9	11	20	-3
855	-9540	-1	-1

**Zadatak 30** Sa standardnog ulaza unosi se neoznačen broj  $N$ . Napisati program koji za uneto  $N$ — iscrtava kvadrat dimenzije  $N$  koji na glavnoj dijagonali ima zvezdice.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	4	7	3
*****	****	*****	***

```

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

```

### 9.2.2 Grupa II

**Zadatak 31** Sa standardnog ulaza unosi se ceo pozitivan broj  $N$ , a potom  $N$  celih brojeva. Naći sumu brojeva koji su deljivi sa 5, a nisu deljivi sa 7. U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	-3	10	6
2 35 5 -175 -20		-5 6 175 -20 -25 -8 42 245 1 6	2205 -1904 2 7 -540 5
-15	-1	-50	-535

**Zadatak 32** Sa standardnog ulaza unose se realni brojevi  $x$  i  $a$  i ceo pozitivan broj  $n$  veći od 0. Napisati program koji za učitane vrednosti  $x$ ,  $a$  i  $n$  izračunava:

$$\underbrace{((\dots(((x+a)^2+a)^2+\dots a)^2)}_n$$

U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3.2	2	2.6	5.4
0.2	1	0.3	7
5	3	3	-2
367940960.000000	101.000000	76.164085	-1

**Zadatak 33** Sa standarnog ulaza unose se neoznačeni celi brojevi  $M$  i  $N$ . Napisati program koji za učitane brojeve  $M$  i  $N$  ispisuje jedan do drugog  $N$  kvadrata čija je svaka strana sastavljena od  $M$  zvezdica. Zvezdice su međusobno razdvoje prazninom. Na primer, za  $M = 5$  i  $N = 3$ :

Primer 1:	Primer 2:
5	3
3	5

```

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

```

```

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

```

Primer 3:	Primer 4:
7	4
2	4

```

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

```

```

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

```

### 9.2.3 Grupa III

**Zadatak 34** Sa standardnog ulaza unose se neoznačeni brojevi N i A, a potom i N celih brojeva. Odrediti koliko ima parova uzastopnih brojeva koji se nalaze na rastojanju A.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	3	10	6
2	19	5	20
2 3 5 1 -1	-30 1 12	-3 6 11 -20 -25 -8 42 37 1 6	1 -19 2 7 27 3
2	0	4	2

**Zadatak 35** Sa standardnog ulaza unosi se ceo pozitivan broj n veći od 0. Napisati program koji računa sledeću vrednost:

$$S = (1 + \frac{1}{2!})(1 + \frac{1}{3!}) \dots (1 + \frac{1}{n!})$$

U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	7	0	10
1.838108	1.841026	-1	1.841077

**Zadatak 36** Sa standardnog ulaza se u citava neparan ceo broj N. Napisati program koji za učitani N ispisuje sledeću figuru:

Primer 1:	Primer 2:	Primer 3:	Primer 4:
5	-3	7	2
* *	-1	* *	-1
* *		* *	
*****		* *	
* *		*****	
* *		* *	
		* *	
		* *	

U slučaju greške ispisati -1.

## 9.3 Završni ispit, januarski rok, 23.01.2014

### 9.3.1 Deo 2, grupa 1

**Zadatak 37** Napisati URM program kojim se izračunava funkcija:  $f(x) = \begin{cases} x - 2, & x \geq 2 \\ 0, & x < 2 \end{cases}$

**Zadatak 38** (3 poena) Sa standardnog ulaza unosi se neoznačen ceo broj. Formirati novi broj koji se dobija izbacivanjem svake druge cifre iz polaznog broja. Cifre se posmatraju sa desna na levo. Rešenje ispisati na izlaz.

**Napomena 1:** Neispravnim se smatra rešenje gde se ispišu cifre, a ne formira broj.

**Napomena 2:** Ne razmatrati slučaj negativnih brojeva.

Primer 1: 12345	Primer 2: 23	Primer 3: 0	Primer 4: 2
135	3	0	2

**Zadatak 39** (4 poena) Sa standardnog ulaza unosi se ceo nenegativan broj  $n$  ( $n \geq 1$ ). Napisati program koji na standardni izlaz ispisuje sve sume oblika  $S = 1 + 8 + 27 + \dots + k^3$  za svako  $k$  ( $0 < k \leq n$ ). U slučaju greške ( $n < 1$ ) ispisati -1;

Primer 1: 4	Primer 2: 10	Primer 3: 0	Primer 4: -3
1 9 36 100	1 9 36 100 225 441 784 1296 2025 3025	-1	-1

**Zadatak 40** (5 poena) Sa standardnog ulaza unosi se broj  $n$  ( $0 < n \leq 200$ ), a potom i elementi kvadratne matrice dimenzije  $n \times n$ . Elementi matrice su celi brojevi. Proveriti da li je uneta matrica ortonormirana i na standardni izlaz ispisati **da** ako jeste ili **ne** ako nije ortonormirana. Matrica je ortonormirana ako je skalarni proizvod svakog para različitih vrsta jednak 0, a skalarni proizvod vrste sa samom sobom 1. U slučaju greške ispisati -1.

**Napomena 1:** Skalarni proizvod vektora  $a = (a_1, a_2, \dots, a_n)$  i  $b = (b_1, b_2, \dots, b_n)$  je  $a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n$ .

**Napomena 2:** Ukoliko program uvek ispisuje **da** ili uvek ispisuje **ne** smatraće se netačnim i poeni se ne mogu osvojiti.

Primer 1: 4	Primer 2: 3	Primer 3: 3	Primer 4: 5
1 0 0 0	1 2 3	2 -2 1	-1 0 0 0 0
0 0 1 0	4 5 6	1 2 2	0 0 0 1 0
0 0 0 1	7 8 9	2 1 -2	0 0 -1 0 0
0 1 0 0			0 1 0 0 0
	ne	ne	0 0 0 0 -1
da			da

**Zadatak 41** (7 poena) Struktura polinom je definisana na sledeći način:

```
#define STEPEN 10
typedef struct
{
    unsigned char stepen;
    short koef[STEPEN+1];
}_polinom;
```

a) Napisati funkciju `_polinom ucitaj(FILE* f)` koja učitava polinom iz datoteke. U funkciji se iz datoteke prvo učitava stepen polinoma, a potom koeficijenti polinoma. Funkcija vraća učitani polinom.

b) Napisati funkciju `void ispis(_polinom p)` koja ispisuje polinom na standardni izlaz u obliku  $k_0 \cdot x^0 + k_1 \cdot x^1 + k_2 \cdot x^2 + \dots + k_n \cdot x^n$ , ukoliko je  $n$  stepen polinoma.

c) Napisati funkciju `_polinom` `izvod(_polinom p)` koja izračunava prvi izvod polinoma `p` i dobijeni polinom vraća kao povratnu vrednost.

U datoteci `polinom.txt` su zapisani polinomi. Prvo je dat broj polinoma (nije ograničeno), a potom u svakom redu jedan polinom. Polinom se zadaje stepenom, a potom i svojim koeficijentima. Napisati program koji za svaki učitani polinom iz datoteke radi sledeće:

a) (3 poena) Ispisuje polinom na standardni izlaz.

b) (4 poena) Izračunava izvod polinoma i ispisuje ga na standardni izlaz, u novom redu.

Između izvoda prethodnog polinoma i sledećeg polinoma napraviti red razmaka (videti test primer). U slučaju greške ispisati `-1`.

Primer 1:  
polinom.txt:  
2  
3 1 0 3 1  
4 7 9 4 0 4

standardni izlaz:  
 $1*x^0 + 0*x^1 + 3*x^2 + 1*x^3$   
 $0*x^0 + 6*x^1 + 3*x^2$   
  
 $7*x^0 + 9*x^1 + 4*x^2 + 0*x^3 + 4*x^4$   
 $9*x^0 + 8*x^1 + 0*x^2 + 16*x^3$

Primer 2:  
polinom.txt:  
3  
5 82 34 11 -2 3 7  
6 11 -9 0 0 2 34 -4  
2 23 100 1

standardni izlaz:  
 $82*x^0 + 34*x^1 + 11*x^2 + -2*x^3 + 3*x^4 + 7*x^5$   
 $34*x^0 + 22*x^1 + -6*x^2 + 12*x^3 + 35*x^4$   
  
 $11*x^0 + -9*x^1 + 0*x^2 + 0*x^3 + 2*x^4 + 34*x^5 + -4*x^6$   
 $-9*x^0 + 0*x^1 + 0*x^2 + 8*x^3 + 170*x^4 + -24*x^5$   
  
 $23*x^0 + 100*x^1 + 1*x^2$   
 $100*x^0 + 2*x^1$

---

Primer 3:  
polinom.txt:  
3  
3 1 1 2 3  
4 1 90 5 7 8  
1 1 2

standardni izlaz:  
 $1*x^0 + 1*x^1 + 2*x^2 + 3*x^3$   
 $1*x^0 + 4*x^1 + 9*x^2$   
  
 $1*x^0 + 90*x^1 + 5*x^2 + 7*x^3 + 8*x^4$   
 $90*x^0 + 10*x^1 + 21*x^2 + 32*x^3$   
  
 $1*x^0 + 2*x^1$   
 $2*x^0$

Primer 4:  
polinom.txt:  
1  
3 7 -5 2 1

standardni izlaz:  
 $7*x^0 + -5*x^1 + 2*x^2 + 1*x^3$   
 $-5*x^0 + 4*x^1 + 3*x^2$

**Zadatak 42** (6 poena) Napisati program koji u datoteku `izlaz.txt` prepisuje sve reči iz datoteke `ulaz.txt` čiji je zbir ascii kodova slova strogo veći od 1000. Reči su odvojene prazninama i nisu duže od 200 karaktera.

Primer 1:  
ulaz.txt:  
Sa standardnog ulaza unosi se neoznaceni

Primer 2:  
ulaz.txt:  
konstruisanje test-prимера sa

ceo broj. Formirati novi broj koji se dobija izbacivanjem svake druge cifre iz polaznog broja.

i dugackim recima kao prestolonaslednik brojevima1234567890

izlaz.txt:  
standardnog izbacivanjem

izlaz.txt:  
konstruisanje test-primera  
prestolonaslednik  
brojevima1234567890

Primer 3:  
ulaz.txt:  
ima jos dugackih reci: predskazanje,  
potom  
nelogicnosti, zanemarivati, odugovlaciti, a ima  
i i malih reci koje su kratke  
predosecaj

Primer 4:  
ulaz.txt:  
i sada jedan kratak primer  
p1: 1234567890  
p2: ABCDEFGHIJ  
p3: abcdefghij

izlaz.txt:  
predskazanje, nelogicnosti,  
zanemarivati, odugovlaciti,  
predosecaj

izlaz.txt:  
abcdefghij

### 9.3.2 Deo 2, grupa 2

**Zadatak 43** Napisati URM program kojim se izracunava funkcija:  $f(x, y) = \begin{cases} 2 * x, & x > y \\ x + y, & \text{inace} \end{cases}$

**Zadatak 44** (3 poena) Sa standardnog ulaza unosi se neoznaceni ceo broj. Formirati novi broj koji se dobija izbacivanjem svake trece cifre iz polaznog broja. Cifre se posmatraju sa desna na levo. Resenje ispisati na izlaz.

**Napomena 1:** Neispravnim se smatra resenje gde se ispišu cifre, a ne formira broj.

**Napomena 2:** Ne razmatrati slucaj negativnih brojeva.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
12345	238	0	20
1245	38	0	20

**Zadatak 45** (4 poena) Sa standardnog ulaza unose dva broja ceo pozitivan ceo broj  $n$  i ceo broj  $m$  ( $n \leq 10, |m| < 10$ ). Napisati program koji na standardni izlaz ispisuje sve sume oblika  $S = m^1 + m^2 + \dots + m^k$  za svako  $k$  ( $0 < k \leq n$ ). U slucaju greške ispisati -1;

Primer 1:	Primer 2:	Primer 3:	Primer 4:
4 2	1 1	0 2	4 15
2 6 14 30	1	-1	-1

**Zadatak 46** (5 poena) Sa standardnog ulaza se unosi broj  $n$  ( $0 < n \leq 10$ ), a potom i elementi kvadratne matrice dimenzije  $n \times n$ . Elementi matrice su celi brojevi. Proveriti da li važi da su zbrojevi elemenata kolona matrice uredjeni u strogo rastućem poretku. **Napomena 1:** Ukoliko program uvek ispisuje da ili uvek ispisuje ne smatraće se netačnim i poeni se ne mogu osvojiti.

Primer 1: 4 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0  ne	Primer 2: 3 1 2 3 4 5 6 7 8 9  da	Primer 3: 3 2 -2 1 1 2 2 2 1 -2  ne	Primer 4: 5 -1 0 2 0 20 0 0 0 10 0 0 0 -1 0 0 0 1 0 0 0 0 0 0 0 -1  da
--	---	---	--

**Zadatak 47** (7 poena) *Struktura polinom je definisana na sledeći način:*

```
#define STEPEN 10
typedef struct
{
unsigned char stepen;
float koef[STEPEN+1];
}_polinom;
```

- Napisati funkciju `_polinom ucitaj(FILE* f)` koja učitava polinom iz datoteke. U funkciji se iz datoteke prvo učitava stepen polinoma, a potom koeficijenti polinoma. Funkcija vraća učitani polinom.
- Napisati funkciju `void ispis(_polinom p)` koja ispisuje polinom na standardni izlaz u obliku  $k_0*x^0 + k_1*x^1 + k_2*x^2 + \dots + k_n*x^n$ , ukoliko je  $n$  stepen polinoma.
- Napisati funkciju `_polinom integral(_polinom p)` koja izračunava integral polinoma  $p$  i dobijeni polinom vraća kao povratnu vrednost.

U datoteci `polinom.txt` su zapisani polinomi. Prvo je dat broj polinoma (nije ograničeno), a potom u svakom redu jedan polinom. Polinom se zadaje stepenom, a potom i svojim koeficijentima. Napisati program koji za svaki učitani polinom iz datoteke radi sledeće:

- (3 poena) Ispisuje polinom na standardni izlaz.
- (4 poena) Izračunava integral polinoma i ispisuje ga na standardni izlaz, u novom redu.

Između integrala prethodnog polinoma i sledećeg polinoma napraviti red razmaka (videti test primer). U slučaju greške ispisati -1. **Napomena 1:** Vrednosti koeficijenata pisati zaokruženo na 4 decimale, inače se rešenje neće prihvatati.

Primer 1:  
polinom.txt:  
2  
3 1 0 3 1  
4 7 9 4 0 4

standardni izlaz:  
1.0000\*x^0 + 0.0000\*x^1 + 3.0000\*x^2 + 1.0000\*x^3  
0.0000\*x^0 + 1.0000\*x^1 + 0.0000\*x^2 + 1.0000\*x^3 + 0.2500\*x^4  
7.0000\*x^0 + 9.0000\*x^1 + 4.0000\*x^2 + 0.0000\*x^3 + 4.0000\*x^4  
0.0000\*x^0 + 7.0000\*x^1 + 4.5000\*x^2 + 1.3333\*x^3 + 0.0000\*x^4 + 0.8000\*x^5

Primer 2:  
polinom.txt:  
1  
3 1 0 4 1

Primer 3:  
polinom.txt:  
0  
standardni izlaz: -

standardni izlaz:

```
1.0000*x^0 + 0.0000*x^1 + 4.0000*x^2 + 1.0000*x^3
0.0000*x^0 + 1.0000*x^1 + 0.0000*x^2 + 1.3333*x^3 + 0.2500*x^4
```

Primer 4:

polinom.txt:

```
3
3 1 0 4 1
2 1 2 3
1 0 1
```

standardni izlaz:

```
1.0000*x^0 + 0.0000*x^1 + 4.0000*x^2 + 1.0000*x^3
0.0000*x^0 + 1.0000*x^1 + 0.0000*x^2 + 1.3333*x^3 + 0.2500*x^4
1.0000*x^0 + 2.0000*x^1 + 3.0000*x^2
0.0000*x^0 + 1.0000*x^1 + 1.0000*x^2 + 1.0000*x^3
0.0000*x^0 + 1.0000*x^1
0.0000*x^0 + 0.0000*x^1 + 0.5000*x^2
```

**Zadatak 48** (6 poena) Napisati program koji u datoteku `izlaz.txt` prepisuje sve reči iz datoteke `ulaz.txt` čiji je zbir ascii kodova slova strogo veći od 1000. Reči su odvojene prazninama i nisu duže od 200 karaktera.

Primer 1:

`ulaz.txt`:

Sa standardnog ulaza unosi se neoznaceni broj. Formirati novi broj koji se dobija izbacivanjem svake druge cifre iz polaznog broja.

`izlaz.txt`:

standardnog izbacivanjem

Primer 2:

`ulaz.txt`:

konstruisanje test-primera sa dugackim recima kao prestolonaslednik brojevima1234567890

`izlaz.txt`:

konstruisanje test-primera prestolonaslednik brojevima1234567890

Primer 3:

`ulaz.txt`:

ima jos dugackih reci: predskazanje, potom nelogicnosti, zanemarivati, odugovlaciti, a ima i i malih reci koje su kratke predosecaj

`izlaz.txt`:

predskazanje, nelogicnosti, zanemarivati, odugovlaciti, predosecaj

Primer 4:

`ulaz.txt`:

i sada jedan kratak primer  
p1: 1234567890  
p2: ABCDEFGHIJ  
p3: abcdefghij

`izlaz.txt`:

abcdefghij

## 9.4 Završni ispit, februarski rok, 07.02.2014

1. Napisati program koji proverava da li su cifre zadatog pozitivnog broja uredjene ciklično. Na primer, cifre broja 73519 su uredjene ciklično jer važi:  $7 \geq 3 \leq 5 \geq 1 \leq 9$



2. Za unetih 5 brojeva, pronaći dva najmanja.
3. Napisati program koji za unetu nisku proverava da li je njome zadat korektan heksadekadni broj. Broj je korektno zadat ako se u njemu javljaju samo cifre i samo mala ili samo velika slova  $a, b, c, d, e$  i  $f$ .
4. Napisati program koji iz datoteke *sabirci.txt* učitava dva cela broja  $n$  i  $m$  ( $10 \leq n, m \leq 20$ ), a zatim i dva niza dužina  $n$  i  $m$  kojima se zadaju dva velika broja svojim ciframa. Napisati program koji izračunava zbir ovih brojeva i rezultat upisuje u datoteku *zbir.txt* u istom formatu: u prvoj liniji dužinu niza, a u drugoj liniji cifre zbira počevši od cifre najveće težine.
5. U datoteci *matrica.txt* je zadata dimenzija kvadratne matrice, a zatim i vrednosti pojedinačnih elemenata. Napisati program koji:
  - učitava matricu iz zadate datoteke
  - proverava da li zbrovi elemenata svih dijagonala matrice paralelnih glavnoj dijagonali čine rastući niz.
6. Napisati program pomoću kojeg se za dati broj  $n$  izračunava  $n$ -ti član niza  $F_n = 3 * F_{n-1} - 2 * F_{n-2} + F_{n-1} * F_{n-2}$  pri čemu je  $F_0 = 1$  i  $F_1 = 1$ . Ne koristiti nizove.
7. **A)** Napisati funkciju `void brojanje(int a[], int brojac[], int N)` čiji su argumenti  $a$  i `brojac` celobrojni nizovi dimenzije  $N$ . Vrednosti elemenata niza  $a$  su između  $0$  i  $N - 1$ . Funkcija izračunava elemente niza `brojac` tako da je `brojac[i]` jednak broju pojavljivanja broja  $i$  u nizu  $a$ .  
**B)** Kažemo da je celobrojni niz  $a$  dimenzije  $N$  permutacija ako sadrži svako  $i$ :  $0 \leq i < N$ . Sastaviti funkciju `int DaLiJePermutacija(int a[], int N)` koja vraća  $1$  ako je niz  $a$  permutacija, a inače  $0$  (koristiti funkciju `brojanje`). Napisati program koji testira rad ove funkcije.
8. Datoteka *Matrice.txt* sadrži dve celobrojne kvadratne matrice. U datoteci su prvo zapisane dimenzije matrica  $n$  i  $m$  ( $n \neq m$ ), a zatim i elementi prvo jedne a zatim i druge matrice. Napisati program koji proverava da li se manja matrica sadrži u većoj. Matrica se sadrži u matrici veće dimenzije ukoliko postoji podmatrica veće matrice identična manjoj matrici tj. ako postoji blok veće matrice dimenzije  $m \times m$  čiji su elementi jednaki elementima manje matrice na odgovarajućim pozicijama. npr. U matrici

```
1 1 1
2 2 2
3 3 3
```

se sadrži matrica

```
1 1
2 2
```

a ne sadrži matrica

```
1 1
2 3
```

9. 6. Struktura `_datum` je definisana na sledeći način:

```
typedef struct Datum
{
int dan;
int mesec;
int godina
}_datum;
```

- A) Napisati funkciju koja određuje da li prvi datum prethodi drugom datumu.
- B) Sastaviti funkciju koja za dati niz datuma vraća datum koji prethodi svim ostalim datumima u nizu.
- C) Napisati program u kome se sa standardnog ulaza učitava niz datuma i ispisuje onaj koji prethodi svim ostalim datumima.

U slučaju greke ispisati -1.

10. Definisati strukturu

```
typedef struct{
    int x;
    int y;
    int z;
} v;
```

kojom se opisuje trodimenzioni vektor. Napisati program koji učitava broj vektora, a zatim i niz vektora i ispisuje na standardnom izlazu koordinate vektora sa najvećom dužinom. Dužina vektora se izračunava po formuli:

$$|v| = \sqrt{x^2 + y^2 + z^2}$$

11. Napisati URM program koji izračunava funkciju:  $f(x, y) = 2 * (x - 1) + y$

12. URM zadatak: Napisati urm program koji izračunava sledeću funkciju:  $f(x, y) = \begin{cases} 3 \cdot x + 1, & x > y \\ y, & \text{inace} \end{cases}$

## 9.5 Završni ispit, junski rok, 10.06.2014

### 9.5.1 URM

Napisati URM program kojim se izračunava funkcija:  $f(x) = \begin{cases} 2x - 3, & x \geq 2 \\ 0, & x < 2 \end{cases}$

### 9.5.2 Deo II

**Zadatak 49** (3 poena) Sa standardnog ulaza se učitavaju tri realna broja tipa float. Na standardni izlaz ispisati (na tri decimale) prvo najmanji, pa najveći i na kraju prosek tri data broja.

Primer 1:	Primer 2:	Primer 3:
1.234 0.111 5.432	0.123 -3.456 9.1	2 0 -6
0.111 5.432 2.259	-3.456 9.100 1.922	-6.000 2.000 -1.333

**Zadatak 50** (4 poena) Sa standardnog ulaza se učitava broj N ( $N \geq 2$ ). Na standardni izlaz ispisati stelicu dimenzije N sačinjenu od zvezdica (pogledati primere). Za  $N < 2$  ispisati -1 i prekinuti izvršavanje programa.

Primer 1:	Primer 2:	Primer 3:
2	3	4
* * * * * *	* * * * * * * * *	* * * * * * * * * * * * * *

**Zadatak 51** (6 poena) *Napisati funkciju*

```
void prepis(char a[][21], int na, char b[][21], int *nb)
```

koja iz niza reči **a** dužine **na** prepisuje u niz **b** reči koje su zapisane samo malim ili samo velikim slovima. Informaciju o dužini niza **b** (broj reči koje zadovoljavaju prethodni uslov) smestiti u promenljivu **nb**.

Napisati program koji sa standardnog ulaza učitava prvo broj reči (strogo veći od nule, manji od 50), a zatim i same reči razdvojene blanko znakom (smatrati da reči koje se unose sa ulaza neće biti duže od 20 karaktera - ovaj uslov ne proveravati). Za slučaj kada je broj reči izvan traženog opsega ispisati -1 i prekinuti izvršavanje programa. Korišćenjem prethodno definisane funkcije `prepis`, odrediti sve reči koje su zapisane samo malim ili samo velikim slovima. Rezultat ispisati na standardni izlaz. Napomena: Ukoliko se pri rešavanju zadatka ne bude koristila funkcija `prepis`, zadatak neće biti pregledan i nosiće nula poena.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
3	4	2	-3
abc ABC aBc	2abc AVF\$ abc AV4	mnB RGa	-1
abc ABC	abc		

**Zadatak 52** (6 poena) *Svaka tačka u ravni se predstavlja dvema svojim realnim koordinatama tipa float.*

- Opisati tačku u ravni strukturom.
- Napisati funkciju koja za datu tačku računa rastojanje (tipa float) od koordinatnog početka.
- Napisati funkciju koja za dati niz tačaka ispisuje (na tri decimale) prosečnu vrednost rastojanja svih tačaka u nizu od koordinatnog početka. Ispisati zatim (na tri decimale) one tačke koje se nalaze na udaljenosti većoj (ili jednako) od ove prosečne vrednosti rastojanja.
- Napisati program koji sa standardnog ulaza učitava broj tačaka  $N$  ( $0 < N < 50$ ), a zatim i tačke zadate svojim koordinatama. Testirati rad prethodno definisane funkcije. Ukoliko se za broj tačaka unese vrednost manja (ili jednaka) od nule, ispisati -1. Ne vršiti dodatne provere.

Primer 1:	Primer 2:	Primer 3:
2	3	1
2 4	1.234 -5.6	2.3 4.5
3 5	2.34 6.7	5.054
5.152	-1.2 -3.4	2.3 4.5
3.000 5.000	5.479	
	1.234 -5.600	
	2.340 6.700	

**Zadatak 53** (6 poena) *U datoteci matrica.txt zadate su dimenzija kvadratne matrice  $n$  ( $0 < n \leq 50$ ), a zatim i vrednosti pojedinačnih elemenata. Ukoliko je  $n$  izvan ovog opsega ispisati -1 i prekinuti izvršavanje programa. Napisati program koji:*

- Učitava matricu iz zadate datoteke i ispisuje je na izlaz. U slučaju greške ispisati -1 i prekinuti izvršavanje programa.
- Ispituje da li su elementi matrice po kolonama, vrstama i dijagonalama (glavnoj i sporednoj) sortirani strogo rastuće. Za svaki od ovih slučajeva redom ispisati 1 ako jesu i 0 ako nisu sortirani - videti primere.

Primer 1: matrica.txt 3 1 2 3 4 5 6 7 8 9  Izlaz: 1 2 3 4 5 6 7 8 9 1 1 1	Primer 2: matrica.txt 2 6 9 4 10  Izlaz: 6 9 4 10 0 1 0	Primer 3: matrica.txt 4 5 5 7 9 6 10 11 13 8 12 14 15 13 15 16 20  Izlaz: 5 5 7 9 6 10 11 13 8 12 14 15 13 15 16 20 1 0 1	Primer 4: matrica.txt 1 5  Izlaz: 5 1 1 1
--	--	--	--

## 9.6 Završni ispit, septembarski rok, 05.09.2014

- Napisati URM program kojim se izračunava funkcija:  $f(x) = \begin{cases} x - 100, & x \geq 100 \\ 0, & x < 100 \end{cases}$
- Sa standardnog ulaza unosi se neoznačen ceo broj. Formirati novi broj koji se dobija sažimanjem – svaka serija istih cifara se zamenjuje jednom cifrom iz serije. Rešenje ispisati na izlaz.  
**Napomena 1:** Neispravnim se smatra rešenje gde se ispišu cifre, a ne formira broj.  
**Napomena 2:** Ne razmatrati slučaj negativnih brojeva.

Primer 1: 11122  12	Primer 2: 123  123	Primer 3: 5555  5	Primer 4: 12233344  1234
------------------------------	-----------------------------	----------------------------	-----------------------------------

- Sa standardnog ulaza unose se dva neoznačena cela broja –  $m$  i  $n$  (oba veća od 0 i  $m \geq n$ ). Na standardni izlaz ispisati sve sume počevši od 1 do  $m$  koje su dužine  $n - s_1 = 1 + \dots + n$ ,  $s_2 = 2 + \dots + (n + 1)$ , ...,  $s_{m-n+1} = (m - n + 1) + \dots + m$ . U slučaju greške ispisati -1.

Primer 1: 10 3  6 9 12 15 18 21 24 27	Primer 2: 12 7  28 35 42 49 56 63	Primer 3: 12 15  -1	Primer 4: 5 5  15
--	--	------------------------------	----------------------------

- Sa standardnog ulaza unose se celi pozitivni brojevi  $m$  i  $n$  koji označavaju broj vrsta i broj kolona matrice. Potom se unose elementi matrice. Nakon unosa elemenata matrice, unose se još dva broja  $p$  i  $k$  ( $p \leq m$ ,  $k \leq n$ ). Na standardni izlaz ispisati sume svih podmatrica (dimenzije  $p \times k$ ) unete matrice. U slučaju greške ispisati -1.

**Napomena 1:** Ne razmatrati slučaj negativnih brojeva.  
**Napomena 2:** Nije bitan redosled kojim se ispisuju sume.

Primer 1: 3 4 1 2 3 4 5 6 7 8 9 10 11 12 3 3	Primer 2: 3 4 1 2 3 4 5 6 7 8 9 10 11 12 2 3	Primer 3: 3 2 1 2 3 4 5 6 7 8	Primer 4: 5 3 1 1 2 5 0 2 7 8 9 1 2 4
---	---	--	--

54 63

24 30 48 54

-1

0 1 1

2 2

7 5 20 19 18 23 4 8

- U datoteci `učenici.txt` dati su podaci o učenicima. U prvom redu datoteke dat je broj učenika (nije ograničeno koliko). Potom u svakom sledećem redu dato je ime učenika (maksimalne dužine 20 karaktera) i 9 ocena (ocene su celi brojevi od 1 do 5). Na standardni izlaz ispisati:

- a) Reč `NEDOVOLJNI:`, a potom imena nedovoljnih učenika. Učenik je nedovoljan ako ima barem jednu jedinicu.
- b) Potom ispisati reč `ODLICNI:`, a potom imena odličnih učenika. Učenik je odličan ako ima prosek ocena veći ili jednak od 4.5.

U slučaju greške (ne postoji datoteka) ispisati -1.

Primer 1:

5

Maja 4 5 2 3 4 4 3 3 4

Nikola 5 4 5 5 5 4 4 5 5

Jasmina 2 2 1 1 2 3 3 1 3

Pera 5 4 5 3 5 5 1 5 5

Pavle 4 3 2 4 3 2 4 3 2

`NEDOVOLJNI:` Jasmina Pera

`ODLICNI:` Nikola

Primer 2:

3

Uros 3 4 2 3 4 2 3 4 4

Nebojsa 4 5 5 5 4 5 5 5 5

Sreten 2 3 2 4 5 4 4 4 2

`NEDOVOLJNI:`

`ODLICNI:` Nebojsa

Primer 3:

6

Vesna 2 4 5 4 4 3 3 2 3

Petar 5 5 5 4 5 5 5 5 5

Kristina 2 3 1 2 3 3 3 3 3

Jovan 4 3 2 1 1 1 3 3 3

Uros 5 5 5 5 4 4 5 5 5

Klara 4 3 2 2 2 3 4 4 4

`NEDOVOLJNI:` Kristina Jovan

`ODLICNI:` Petar Uros

Primer 4:

4

Mirko 2 3 4 4 4 3 3 3 4

Mihailo 4 5 4 4 4 4 4 4 4

Dragan 2 3 4 2 3 3 3 4 4

Ilija 5 5 4 3 2 3 4 4 2

`NEDOVOLJNI:`

`ODLICNI:`

- Sa standardnog ulaza unose se celi pozitivni brojevi  $k$  i  $n$ . Napisati program koji iz datoteke `ulaz.txt` prepisuje svaku  $k$ -tu reč u kojoj je izbačeno svako  $n$ -to slovo u datoteku `izlaz.txt`. Reč je niz karaktera koji nema beline. Maksimalna dužina reči u tekstu nije unapred poznata.

U slučaju greške (negativnih vrednosti za  $k$  i  $n$  ili problema pri otvaranju datoteke) ispisati -1.

Primer 1:

3 2

`ulaz.txt:`

Sa standardnog ulaza unosi se neoznaceni ceo broj. Formirati novi broj koji se dobija izbacivanjem svake druge cifre iz polaznog broja.

Primer 2:

4 5

`ulaz.txt:`

konstruisanje test-primera sa i dugackim recima kao prestolonaslednik brojevima1234567890

izlaz.txt:  
uaa noncn Friai kj ibcvne cfe boa

izlaz.txt:  
i presolonsledik

---

Primer 3:  
5 3  
ulaz.txt:  
ima jos dugackih reci: predskazanje,  
potom  
nelogicnosti, zanemarivati, odugovlaciti, a ima  
i i malih reci koje su kratke  
predosecaj

izlaz.txt:  
prdsaznj a rei

Primer 4:  
2 4  
ulaz.txt:  
i sada jedan kratak primer  
p1: 1234567890  
p2: ABCDEFGHIJ  
programiranje3: abcdefghij

izlaz.txt:  
sad kraak p1: p2: proramrane3:

# Glava 10

## Akademska godina 2014/2015

### 10.1 Test 2, 29.11.2014.

#### 10.1.1 Grupa I

1. Napisati URM program koji izračunava funkciju  
 $f(x, y, z) = 2 * (x + 1) + y + z$ .
2. Napisati program koji za uneti pozitivan trocifren broj formira i ispisuje najmanji broj koji se sastoji iz istih cifara. U slučaju pogrešnog unosa ispisuje se -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
532	505	400	-102	1234	8
235	55	4	-1	-1	-1

3. Napisati program koji za uneti ceo broj  $n > 0$ , iscrtava figuru u obliku zadanom test primerima. U slučaju pogrešnog unosa ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
4	3	2	1	0
*--*--*	*--*	*--*	*	-1
*--*	*--*	*		
*--*	*			
*				

#### 10.1.2 Grupa II

1. Napisati URM program koji izračunava funkciju:

$$f(x, y) = \begin{cases} 2 * y, & x < y, \\ x - y, & x \geq y. \end{cases}$$

2. Napisati program koji za uneti pozitivan trocifren broj formira i ispisuje broj koji se dobija kada se sve neparne cifre broja (cifre koje imaju neparnu vrednost) umanje za 1. U slučaju pogrešnog unosa ispisuje se -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
532	123	111	-102	1234	8
422	22	0	-1	-1	-1

3. Napisati program koji za uneti ceo broj  $n > 0$ , iscrtava jelkicu u obliku zadatom test primerima.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
4	2	1	0	-3
*	*	*	-1	-1
*A*	*A*			
*A*A*				
*A*A*A*				

### 10.1.3 Grupa III

1. Napisati URM program koji izračunava funkciju:

$$f(x, y) = \begin{cases} 2 * x - y, & 2 * x \geq y, \\ 0, & 2 * x < y. \end{cases}$$

2. Sa standardnog ulaza se unosi četvorocifren ceo broj  $x$ . Ako su cifre unetog broja strogo rasutuće ( $c1 < c2 < c3 < c4$ ) ili strogo opadajuće ( $c1 > c2 > c3 > c4$ ) ispisati njihov zbir, ako su alternirajuće ( $c1 > c2 < c3 > c4$  ili  $c1 < c2 > c3 < c4$ ) ispisati njihov proizvod, inače ispisati uneti broj (pretpostavka je da su  $c1, c2, c3$  i  $c4$  cifre broja  $x$ ). U slučaju pogrešnog unosa ispisuje se  $-1$ .

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:	Primer 7:
1234	9642	1538	8251	1223	1563	123
10	21	120	80	1223	1563	-1

3. Napisati program koji za uneti ceo broj  $n > 0$ , iscrtava zvezdice u obliku zadatom test primerima.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
6	3	4	2	1	-2
*	*	*	*	*	-1
***	***	***	***		
* * *	* * *	* * *	*		
* * *	***	* * *			
* * *	*	* * *			
* * *		***			
* * *		*			
* * *					
***					
*					

## 10.2 Test 4, 13.01.2015.

### 10.2.1 Grupa I

1. Sa standardnog ulaza se učitava celobrojni niz  $a$  tako što se prvo učitava dimenzija niza ( $0 < n \leq 50$ ) a zatim i elementi niza. Nakon toga se učitava ceo broj  $m$ . Ispisati na standardni izlaz (ako postoji) indeks  $i$  niza  $a$  (prvi) takav da je zbir članova niza sa indeksima od  $i$  do  $n-1$  (odnosno do kraja niza) jednak zadatom broju  $m$ . Ako takav indeks ne postoji ispisati  $-1$ .



Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
5	4	3	1	1	-2
1 2 3 4 5	2 0 3 -5	1 2 3	5	5	-1
12	-2	9	5	6	
2	1	-1	0	-1	

2. Napisati funkciju `int poredi(char p[], char d[])` koja vraća  $-1$  ukoliko je  $p < d$ ,  $0$  ukoliko je  $p == d$  a  $1$  ako je  $p > d$ , pri čemu su  $p$  i  $d$  dva velika cela neoznačena broja zadata NISKOM svojih cifara (maksimalan broj cifara je 50). Na primer, broj 123456789 je prikazan kao niska "123456789". Smatrati da su brojevi uneti u ispravnom formatu (ne ispitivati mogućnost pogrešnog unosa). Sa standardnog ulaza učitati dva velika cela neoznačena broja kao niske cifara i na standardni izlaz ispisati onaj koji ima manju vrednost (ako su jednaki onda ispisati bilo koji). Koristiti funkciju `poredi()`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
1234567891011	123456789101111	1234567891011	2234567891011	1234567891011
123456789101111	1234567891011	1234567891011	1234567891011	1284567891011
1234567891011	1234567891011	1234567891011	1234567891011	1234567891011

3. Sa standardnog ulaza se učitavaju celi brojevi  $n$  i  $m$  ( $0 < n \leq 50$ ,  $0 < m \leq 50$ ), a zatim i matrica dimenzije  $n \times m$ . Elementi matrice su celi brojevi. Nakon toga se učitava broj  $k$  ( $0 < k \leq 50$ ) a zatim se učitava niz dužine  $k$ . Elementi niza su takođe celi brojevi. Odrediti proizvod unesenog vektora (niza) i matrice i dobijeni rezultat ispisati na standardni izlaz. Ukoliko množenje nije moguće ispisati  $-1$ .

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
2 3	3 3	2 2	1 1	0 2	1 1
1 2 3	-1 -2 3	1 2	5	-1	5
4 5 6	-3 -3 -3	3 4	1		0
2	1 2 3	3	2		-1
7 8	3	1 2 3	10		
39 54 69	1 1 1	-1			
	-3 -3 3				

## 10.2.2 Grupa II

1. Sa standardnog ulaza se učitava celobrojni niz tako što se prvo učitava dimenzija niza ( $1 < n \leq 50$ ) a zatim i elementi niza. Ispisati niz koji se dobija kada se svaki element ulaznog niza zameni zbirom svojih suseda (ne mora se formirati novi niz).

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
5	3	2	1	0	-2
1 2 3 4 5	-1 2 -3	5 3	-1	-1	-1
2 4 6 8 4	2 -4 2	3 5			

2. Napisati funkciju: `void ciklicno_pomeri(char s1[], char s2[])` koja izračunava nisku `s2` tako što nisku `s1` ciklično pomera za jedno mesto udesno. Na primer, ako je niska `s1 = "ABCDE"` onda niska `s2` treba da bude "EABCD".

Napisati program koji učitava sa standardnog ulaza nisku  $s$  sa ne više od 80 karaktera (bez terminalne nule) i prirodan broj  $k$  ( $k \geq 0$ ). Na standardni izlaz ispisati nisku koja je dobijena od niske  $s$  cikličnim pomeranjem za  $k$  mesta udesno (koristiti funkciju `ciklicno_pomeri()`). U slučaju greške (pogrešnog unosa broja  $k$ ) ispisati  $-1$ .

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
ABCDE	aBCDe	ABCDE	ABCDE	a
3	7	0	-2	3
CDEAB	DeaBC	ABCDE	-1	a

3. Napisati funkciju `void sumiraj(int A[][50], int m, int n, int B[], int C[])` koja u niz B upisuje sume vrsta, a u niz C sume kolona matrice A. Dakle, i-ti element niza B će biti suma svih elemenata i-te vrste matrice A, a i-ti element niza C će biti suma svih elemenata i-te kolone matrice A.

Sa standardnog ulaza se učitavaju celi neoznačeni brojevi  $n$  i  $m$  ( $0 < n \leq 50$ ,  $0 < m \leq 50$ ), a zatim i matrica dimenzije  $n \times m$ . Elementi matrice su celi brojevi. Testirati funkciju `sumiraj()` i rezultujuće nizove B i C ispisati na standardni izlaz. U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
3 3	2 3	1 1	1 0	-1 1
1 2 3	1 2 3	5	-1	-1
4 5 6	4 5 6	5		
7 8 9	6 15	5		
6 15 24	5 7 9			
12 15 18				

### 10.2.3 Grupa III

1. Napisati funkciju `int upletiti(int A[], int an, int B[], int bn, int C[])` koja od dva prosleđena niza A i B formira treći niz C takav da su na parnim pozicijama u nizu elementi niza A, a na neparnim pozicijama elementi niza B. Niz C treba da bude dužine  $2 * \min(an, bn)$ . Sa standardnog ulaza učitati dva celobrojna niza A i B tako što se za svaki prvo učitava dimenzija niza ( $0 < n \leq 50$ ) a zatim i elementi niza. Na standardni izlaz ispisati sadržaj niza C dobijenog pozivom funkcije `upletiti()`. U slučaju greške ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
3	2	1	0	2
1 2 3	1 2	-5	-1	1 2
3	3	4		-2
4 5 6	3 4 5	1 2 3 4		-1
1 4 2 5 3 6	1 3 2 4	-5 1		

2. Napisati program u kom korisnik sa standardnog ulaza unosi reči sve dok ne unese reč KRAJ, pri čemu je svaka reč kraća od 20 karaktera (ne računajući terminirajuću nulu). Reč je bilo koji niz karaktera koji ne sadrži razmak. Na standardni izlaz ispisati onu reč koja sadrži najveći broj cifara a -1 ako takve nema. Ako ima više reči sa istim najvećim brojem cifara, ispisati onu koja se javila prva.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
prva 2-ga	prva druga	1, 2, 3,	11abc 1a2b3c4	1 12 kraj
333-a KRAJ	treca KRAJ	4, 5, KRAJ	\$\$s23 5! KRAJ	123 KRAJ
333-a	-1	1,	1a2b3c4	123

3. Napisati funkciju `void sumirajD(int A[][50], int m, int n, int D[])` koja popunjava niz D tako da je nulti član suma elemenata glavne dijagonale kvadratne matrice A, prvi suma elemenata na prvoj donjoj dijagonalnoj paraleli (nju čine elementi odmah ispod glavne dijagonale), drugi element suma druge donje dijagonalne paralele, itd.

Sa standardnog ulaza se učitava ceo broj  $n$  ( $0 < n \leq 50$ ), a zatim i kvadratna matrica dimenzije  $n \times n$ . Elementi matrice su celi brojevi. Testirati funkciju `sumirajD()` i rezultujući niz `D` ispisati na standardni izlaz. U slučaju greške ispisati `-1`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
2 2	3 3	1 1	1 1	1 -2	0 1
1 2	1 -2 3	5	-2	-1	-1
3 -4	4 5 6	5	-2		
-3 3	7 8 9				
	15 12 7				

### 10.3 I smer, Programiranje 1 2014/2015, završni ispit, januar 2015, grupa I

Na *Desktop*-u napraviti direktorijum čije je ime u formatu `InicijaliAsistenta_ImeIPrezime_BrojIndeksa_1`. Na primer, `AZ_PeraPeric_mi14231_1`. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa `1.c`, `2.c`, `3.c` i `4.c`.

**Zadatak 54** (4 poena) Sa standardnog ulaza unose se celi neoznačeni brojevi sve do pojave broja 0. Napisati program koji ispisuje brojeve koji u svom zapisu imaju barem dve pojave cifre 5. U slučajevima kada nema takvih brojeva ispisati `-1`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
23 5052 5553 82 0	89 2 0	90 855 25 15 0	0	555 55 5555 0
5052 5553	-1	855	-1	555 55 5555

**Zadatak 55** (5 poena) Za reč `s` kažemo da zadovoljava uslov ukoliko se sastoji samo od parnih cifara, samo od neparnih cifara, od serije neparnih cifara za kojom sledi serija parnih cifara ili od serije parnih cifara za kojom sledi serija neparnih cifara. Za reč `s` kažemo da ne zadovoljava uslov ukoliko sadrži karakter koji nije cifra ili ako nije u gore opisanom formatu. Napisati funkciju

```
int uslov(char s[])
```

koja vraća vrednost 1 ukoliko reč `s` zadovoljava uslov ili 0 ukoliko reč `s` ne zadovoljava uslov. U glavnom programu učitati dve reči maksimalne dužine 20 karaktera koje ne sadrže beline i ispisati rezultat rada funkcije.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
24613 135784	246 3894	44af 1*3	7924 7481	164 796613
1 1	1 0	0 0	1 0	1 0

**Zadatak 56** (7 poena) Sa standardnog ulaza učitava se ceo broj  $n$  ( $2 \leq n \leq 100$ ), a zatim i kvadratna matrica dimenzije  $n$  koja predstavlja tabelu fudbalskog šampionata. Element matrice  $(i, j)$  ima vrednost 0 ako je ekipa  $i$  izgubila od ekipe  $j$ , vrednost 1 ako je rezultat nerešeno ili se element nalazi na glavnoj dijagonali i vrednost 2 ako je ekipa  $i$  pobedila ekipu  $j$ . Na standardnom izlazu ispisati:

(a) (3 poena) Pročitanu matricu.

(b) (4 poena) Broj ekipa koje imaju više pobeda nego poraza.

U slučaju unosa pogrešne dimenzije ispisati `-1` i prekinuti izvršavanje programa. Može se pretpostaviti da je matrica korektno zadata i da sadrži samo vrednosti 0, 1 i 2.

Primer 1: 3 1 2 2 0 1 1 0 1 1	Primer 2: 1	Primer 3: 3 1 2 0 0 1 1 2 1 1	Primer 4: 2 1 2 0 1	Primer 5: 2 1 1 1 1	Primer 6: 4 1 0 0 2 2 1 2 0 2 0 1 1 0 2 1 1
1 2 2 0 1 1 0 1 1 1	-1	1 2 0 0 1 1 2 1 1 1	1 2 0 1 1	1 1 1 1 0	1 0 0 2 2 1 2 0 2 0 1 1 0 2 1 1 1

**Zadatak 57 (9 poena)**

(a) (1 poen) Definirati strukturu `Tacka` koja sadrži polja `ime`, `x` i `y`. Polje `ime` je niska maksimalne dužine 10 karaktera i predstavlja ime tačke, dok su polja `x` i `y` celobrojne vrednosti koje predstavljaju koordinate tačke.

(b) (2 poena) Definirati funkciju

float d(Tacka A, Tacka B)

kojom se računa rastojanje između tačaka A i B. Rastojanje između tačaka čije su koordinate  $(x_a, y_a)$  i  $(x_b, y_b)$  se računa po formuli

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

(c) (6 poena) Napisati program koji učitava broj tačaka  $n$  ( $4 \leq n \leq 50$ ), a zatim i niz tačaka dužine  $n$  kojim se opisuje konveksan mnogougao. Pretpostaviti da se tačke unose redom, u smeru kretanja kazaljke na satu. Na standardnom izlazu ispisati ime i dužinu najduže dijagonale. Dijagonala je svaka duž koja spaja dve nesusedne tačke. U slučaju unosa pogrešnog broja tačaka ispisati -1 i prekinuti izvršavanje programa. U slučaju većeg broja najdužih dijagonala, ispisati prvu takvu. Dužinu dijagonale je potrebno ispisati na dve decimale.

Napomena: Može doći do odstupanja rezultata na poslednjoj decimali, npr. na nekim sistemima se umesto rezultata 14.87 može dobiti rezultat 14.86 ili 14.88.

Primer 1: 4 A 0 0 B 0 4 C 4 4 D 4 0	Primer 2: 3	Primer 3: 5 T1 2 -1 T2 0 -5 T3 -3 -2 T4 -3 2 T5 0 6	Primer 4: 4 X 4 9 Y 0 14 Z -1 -5 U -3 5	Primer 5: 53
AC 5.66	-1	T2T5 11.00	XZ 14.87	-1

## 10.4 I smer, Programiranje 1 2014/2015, završni ispit, januar 2015, grupa II

Na *Desktop*-u napraviti direktorijum čije je ime u formatu `InicijaliAsistenta.ImeIPrezime.BrojIndeksa.2`. Na primer, `AZ_PeraPeric.mi14231.2`. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa `1.c`, `2.c`, `3.c` i `4.c`.

**Zadatak 58** (4 poena) Sa standardnog ulaza unose se celi neoznačeni brojevi sve do pojave broja 0. Napisati program koji ispisuje brojeve čiji je zbir cifara veći od 10. U slučajevima kada nema takvih brojeva ispisati -1.

Primer 1: 75 5052 1012 39 0	Primer 2: 43 2 0	Primer 3: 87 37 777 11 0	Primer 4: 0	Primer 5: 96 15 7177 46 0
75 5052 39	-1	87 777	-1	96 7177

**Zadatak 59** (5 poena) Napisati funkciju

```
int uslov(char s[])
```

koja vraća vrednost 1 ukoliko su u zapisu reči **s** naizmenično zapisana samo mala i velika slova ili 0 ukoliko ovaj uslov nije ispunjen. U glavnom programu učitati dve reči maksimalne dužine 20 karaktera koje ne sadrže beline i ispisati rezultat rada funkcije.

Primer 1: zImA vEtAr	Primer 2: IsPiT JaNuar	Primer 3: tEsT3 1*3	Primer 4: AbCd aB_3D	Primer 5: 234 aBcD
1 1	1 0	0 0	1 0	0 1

**Zadatak 60** (7 poena) Sa standardnog ulaza učitavaju se celi brojevi  $n$  i  $m$  ( $1 \leq n, m \leq 100$ ), a zatim i matrica dimenzije  $n \times m$  koja predstavlja tabelu utisaka. Element matrice  $(i, j)$  ima vrednost 1 ako je korisnik  $i$  zadovoljan uslugom  $j$ , vrednost -1 ako korisnik  $i$  nije zadovoljan uslugom  $j$  i vrednost 0 ako je korisnik indiferentan. Na standardnom izlazu ispisati:

(a) (3 poena) Pročitanu matricu.

(b) (4 poena) Broj korisnika koji imaju više negativnih nego pozitivnih utisaka.

U slučaju unosa pogrešnih dimenzije ispisati -1 i prekinuti izvršavanje programa. Može se pretpostaviti da je matrica korektno zadata i da sadrži samo vrednosti -1, 0 i 1.

Primer 1: 2 3 1 -1 -1 0 0 1	Primer 2: 1 1 0	Primer 3: 3 4 1 1 1 0 1 -1 -1 0 0 -1 -1 -1	Primer 4: 3 2 0 1 1 1 0 -1	Primer 5: -2 8	Primer 6: 2 4 1 0 0 1 1 -1 -1 -1
1 -1 -1 0 0 1 1	0 0	1 1 1 0 1 -1 -1 0 0 -1 -1 -1 2	0 1 1 1 0 -1 1	-1	1 -1 -1 1 1 -1 -1 -1 1

**Zadatak 61** (9 poena)

(a) (1 poen) Definisati strukturu **Grad** koja sadrži polja **ime**, **x** i **y**. Polje **ime** je niska maksimalne dužine 10 karaktera i predstavlja ime grada, dok su polja **x** i **y** celobrojne vrednosti koje predstavljaju geokordinate grada.

(b) (2 poena) Definisati funkciju

```
float r(Grad g1, Grad g2)
```

kojom se računa rastojanje između gradova  $g_1$  i  $g_2$ . Rastojanje između gradova čije su koordinate  $(x_1, y_1)$  i  $(x_2, y_2)$  se računa po formuli

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- (c) (6 poena) Napisati program koji učitava broj gradova  $n$  ( $2 \leq n \leq 50$ ), a zatim i niz gradova dužine  $n$ . Na standardnom izlazu ispisati imena gradova koji su najudaljeniji i rastojanje između njih. U slučaju unosa pogrešnog broja gradova ispisati -1 i prekinuti izvršavanje programa. U slučaju većeg broja parova gradova sa istim rastojanjem, ispisati prvi takav. Rastojanje između gradova je potrebno ispisati na dve decimale.

Napomena: Može doći do odstupanja rezultata na poslednjoj decimali, npr. na nekim sistemima se umesto rezultata 19.03 može dobiti rezultat 19.04 ili 19.02.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
4	5	4	2	-3
Beograd 0 0	Negotin 2 -1	Zrenjanin 4 9	Beograd 0 0	
Subotica 0 4	Uzice 0 -5	Sofija 0 14	Zagreb -10 0	
Vrsac 4 4	Kraljevo -3 -2	Nis -1 -5		
Kovin 4 0	Sabac -3 2	Ljubljana -3 5		
	Palic 0 6			
Beograd Vrsac	Uzice Palic	Sofija Nis	Beograd Zagreb	-1
5.66	11.00	19.03	10.00	

## 10.5 Programiranje 1, I smer, završni ispit, februar 2015.

1. Sa standarog ulaza se učitava neoznačen ceo broj. Izbaciti sve neparne cifre broja (cifre koje kao vrednost imaju neparan broj). Rezultat ispisati na standardni izlaz.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
12345	23456	206	135	1010	3020
24	246	206	0	0	20

2. Napisati funkciju:

```
int maxRastojanje(char s[], char c)
```

koja izračunava maksimalno rastojanje između dve pojave karaktera  $c$  u okviru niske  $s$ . Ukoliko se karakter  $c$  u niski  $s$  pojavljuje samo jednom ili ni jednom, funkcija treba da vrati -1. Ne razlikovati mala i velika slova.

Napisati program koji sa standardnog ulaza učitava nisku  $s$  (koja ne sadrži beline i koja nije duža od 20 karaktera) i karakter  $c$ . Na standardni izlaz ispisati maksimalno rastojanje između dve pojave karaktera  $c$  u okviru niske  $s$  (pri tome koristiti prethodno definisanu funkciju).

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
Banana a	Programiranje R	cool o	Kruska K	Mango n	Jabuka z
3	6	0	3	-1	-1

3. Sa standardnog ulaza se učitava dimenzija celobrojne kvadratne matrice  $n$  ( $0 < n \leq 50$ ), a zatim i njeni elementi. Napisati program koji u učitanj matrici  $A$  određuje maksimalni element na glavnoj dijagonali a zatim razmenjuje vrstu i kolonu koje se seku u tom elementu. Ukoliko ima više istih maksimalnih elemenata na glavnoj dijagonali, izabрати prvi među njima ( $A[i][i]$  koji ima najmanju vrednost  $i$ ). Rezultujuću matricu ispisati na standardni izlaz.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
3	3	1	2	0	51
1 2 3	1 2 9	5	1 2	-1	-1
4 5 6	4 6 5		3 4		
7 8 9	2 3 6	5			
			1 3		
1 2 7	1 4 9		2 4		
4 5 8	2 6 3				
3 6 9	2 5 6				

4. Definirati strukturu `Osoba` kojom se opisuje jedan unos u imenik: unos sadrži nisku maksimalne dužine 20 karaktera koja predstavlja ime osobe, nisku maksimalne dužine 30 karaktera koja predstavlja prezime osobe i nisku maksimalne dužine 50 karaktera koja predstavlja email adresu osobe.

Napisati program koji učitava ceo broj  $n$  ( $0 < n \leq 50$ ) a zatim podatke o  $n$  osoba. Na standardni izlaz ispisati imena i prezimena svih osoba koje imaju gmail adresu (čija se email adresa završava sa `@gmail.com`). Ukoliko ne postoji ni jedna takva osoba ispisati `-1` i prekinuti izvršavanje programa. Takođe, u slučaju unosa pogrešne dimenzije  $n$  ispisati `-1` i prekinuti izvršavanje programa.

Primer 1:	Primer 2:
3	2
Dusko Dugousko dusko@yahoo.com	Homer Simpson homer@crtaci.com
Pink Panter panter@gmail.com	Mardz Simpson mardz@crtaci.com
Pera Detlic pd@gmail.com	-1

Pink Panter  
Pera Detlic

Primer 3:	Primer 4:
0	51
-1	-1

## 10.6 I smer, Programiranje 1 2014/2015, završni ispit, jun 2015

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsistenta\_ImeIPrezime\_BrojIndeksa**. Na primer, **AZ\_PeraPeric\_mi14231**. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa **1.c**, **2.c**, **3.c** i **4.c**.

**Zadatak 62** (4 poena) Sa standardnog ulaza unose se celi neoznačeni brojevi sve do pojave broja 0. Napisati program koji računa zbir neparnih cifara unetih broja. U slučajevima kada nema takvih brojeva ispisati 0.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
23 5052 5553 82 0	88 2 0	90 855 25 15 0	0	555 55 5555 0
31	0	30	0	45

**Zadatak 63** (5 poena) Za datu reč `s1` kažemo da je ciklična u dužini 3 ako se u njoj stalno ponavljaju tri karaktera u istom redosledu. Recimo, `abcabcabcabc` jeste ciklična, a reč `abcd` nije. Napisati funkciju

```
int ciklicna(char s[])
```

koja vraća vrednost 1 ukoliko reč **s** jeste ciklična u dužini 3 ili 0 ukoliko reč **s** nije ciklična. U glavnom programu učitati reč maksimalne dužine 20 karaktera koje ne sadrže beline i ispisati 0 ukoliko reč nije ciklična ili ispisati karaktere koje čine ciklus ako reč jeste ciklična.

Primer 1: abcabcabcabc	Primer 2: abcd	Primer 3: abcdabcd	Primer 4: 09A09A09A09A09A	Primer 5: AcDAcDAc
abc	0	0	09A	0

**Zadatak 64** (7 poena) Sa standardnog ulaza učitava se ceo broj **n** ( $2 \leq n \leq 100$ ), a zatim i kvadratna matrica dimenzije **n**. Potom se sa standardnog ulaza učitava ceo neoznačen broj **k**. Na standardnom izlazu ispisati:

(a) (3 poena) Pročitanu matricu.

(b) (4 poena) Sumu donje **k**-te dijagonale matrice. Pri tome, donje **k**-ta dijagonala matrice je paralelna glavnoj dijagonali, i nalazi se ispod nje u matrice. Glavna dijagonala je prva dijagonala matrice, donja paralelna sa njom je druga, sledeća paralelna je treća itd.

Primer 1: 3 1 2 2 0 1 1 0 1 1 1	Primer 2: 1 5  0	Primer 3: 3 1 2 0 0 1 1 2 1 1 2	Primer 4: 2 1 2 0 1 1 1 2 0 1 2	Primer 5: 4 1 0 0 2 2 1 2 0 2 0 1 1 0 2 1 1 3 1 0 0 2 2 1 2 0 2 0 1 1 0 2 1 1 4	Primer 6: 5 1 2 3 4 5 6 0 2 3 1 8 2 1 0 2 4 5 0 1 1 3 9 1 0 0 14
--	------------------------------	--	--	--	---



**Zadatak 65** (9 poena)

(a) Definisati strukturu `Trougao` koja sadrži polja `ime`, `a1`, `a2`, `b1`, `b2` i `c1`, `c2`. Polje `ime` je niska maksimalne dužine 10 karaktera i predstavlja ime trougla. Polja `a1` i `a2` su koordinate prve tačke trougla, polja `b1` i `b2` su koordinate druge tačke trougla, a polja `c1` i `c2` su koordinate treće tačke trougla.

(b) Definisati funkciju

```
int jednakokraki(Trougao A)
```

kojom se računa da li je dati trougao jednakokrak. Jednakokraki trougao ima dve stranice jednake dužine. Rastojanje između tačaka čije su koordinate `(xa, ya)` i `(xb, yb)` se računa po formuli

$$d = \sqrt{(xa - xb)^2 + (ya - yb)^2}$$

(c) Definisati funkciju

```
float površina(Trougao A)
```

koja vraća površinu trougla `A`. Površinu trougla računati po formulu  $\sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}$  pri čemu su `a`, `b` i `c` dužine stranica trougla, a  $s = \frac{a+b+c}{2}$ .

(d) Napisati program koji učitava broj trouglova `n` ( $1 \leq n \leq 50$ ), a zatim i niz trouglova dužine `n`. Na standardni izlaz ispisati imena onih trouglova koji su jednakokraki. Potom ispisati ime onog jednakokrakog trougla koji ima najveću površinu među svim jednakokrakim trouglovima.

Primer 1:

Primer 2:

Primer 3:

Primer 4:

Primer 5:

# Glava 11

## Akademska godina 2015/2016

### 11.1 Programiranje 1, kolokvijum, 10.11.2016.

#### 11.1.1 Grupa I

1. Napisati URM program koji izračunava funkciju:

$$f(x, y) = \begin{cases} 2x - y & 2x \geq y \\ 3y & \text{inače} \end{cases}$$

2. Sa standardnog ulaza unose se jedan karakter (**p** ili **n**) i dva pozitivna trocifrena broja. Na osnovu vrednosti unetog karaktera izračunati i ispisati na standardni izlaz:

**p** - zbir cifara na parnim pozicijama unetih brojeva

**n** - zbir cifara na neparnim pozicijama unetih brojeva

Cifre se broje sa desne strane, tako da cifri jedinice odgovara pozicija 1.

U slučaju greške ( ukoliko karakter nije p ili n ili nisu uneti pozitivni trocifreni brojevi )ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
p 235 654	n 567 101	A 432 543	p 102 1234
8	14	-1	-1

3. Sa standardnog ulaza učitava se pozitivan ceo broj  $i$  i ceo broj  $i$  ( $1 \leq i$ ). Na standardni izlaz ispisati broj koji se dobija kada se ukloni  $i$ -ta cifra broja. Cifre se broje sa desne strane, tako da cifri jedinice odgovara pozicija 1. Neispravan ulaz je kada se unose negativan broj ili negativna vrednost ili nula za  $i$  i u tom slučaju na standardni izlaz ispisati -1. Ukoliko broj nema  $i$ -tu cifru broj ostaje nepromenjen.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
35243 2	-14423 1	1234 5	523156 6
3523	-1	1234	23156

## 11.1.2 Grupa II

1. Napisati URM program koji izračunava funkciju:

$$f(x, y, z) = 4x + 2y + 3z$$

2. Korisnik unosi 7 karaktera koji predstavljaju indeks studenta koji je oblika OOGBBBB. OO je oznaka smera i može biti mi, ma, mr, ms, mm, mv. GG je oznaka godine upisa. BBB je oznaka broja koji može biti jednocifren, trocifren ili dvocifren sa vodećim nulama. Na osnovu ovih podataka na standardni izlaz ispisati ime smera kome student pripada i indeks u obliku broj/godina. U slučaju greške ( ukoliko OO kao oznaka smera nije ispravna ili ostali karakteri nisu brojevi ) ispisati -1. Nazivi smerova su: mi - informatika, ma - astronomija, mr - racunarstvo i informatika, ms - statistika, mm - teorijska matematika, mp - primenjena matematika

Primer 1:	Primer 2:
mi11275	mm98005
informatika 275/2011	teorijska matematika 5/1998

Primer 3:	Primer 4:
mo23112	ms12001
-1	statistika 1/2012

3. Državna lutrija došla je na ideju o novoj igri na sreću. Ova igra na sreću igra se tako što se izvuče jedan broj od 1000 do 9999, Nagrada koja se dobija ako ste pogodili izvučen broj je proizvod njegovih parnih cifara i samog broja. Vaš zadatak je da na osnovu izučenog broja izračunate nagradu koja se dobija. Kao ulaz sigurno ćete dobiti ispravan broj. Ako broj nema parnih cifara, nagrada je sam taj broj. Na standardni izlaz ispišite nagradu.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:	Primer 6:
1321	3284	1111	2222	6031	4321
2642	210176	1111	35552	0	34568

## 11.1.3 Grupa III

1. Napisati URM program koji izračunava funkciju:

$$f(x, y, z) = \begin{cases} 2 \cdot x + 2 \cdot y & x \leq z \\ z + 3 & \text{inače} \end{cases}$$

2. Napisati C program koji sa standardnog ulaza učitava 4 velika slova abedec i nenegativan ceo broj k. Program na standardni izlaz ispisuje 4 karaktera koji se dobijaju cikličkim pomeranjem (u okviru karakterske tabele) unetih karaktera za k mesta unapred. Na primer, karakter A pomeren za 4 mesta unapred postaje E dok karakter Z pomeren za 3 mesta unapred postaje C. U slučaju neispravnog ulaza ispisati -1. Ulaz se smatra neispravnim ako neki od unetih karaktera ne predstavlja veliko slovo abedec ili ako je broj k negativan, pretpostaviti da se na ulazu uvek zadaje tačno četiri karaktera.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5
BABA 3	DEDA 26	ZABC 53	PERA -2	abcd
EDED	DEDA	ABCD	-1	-1

3. Napisati C program koji sa standardnog ulaza učitava dva četvorocifrena, pozitivna, cela broja i proverava da li je broj koji se dobija učešljavanjem unetih brojeva palindrom. Ako uneti brojevi imaju cifre

$a_1 a_2 a_3 a_4$  i  $b_1 b_2 b_3 b_4$  tada su cifre učešljanog broja  $a_1 b_1 a_2 b_2 a_3 b_3 a_4 b_4$ . Broj je palindrom ako se čita isto sa obe strane. Ukoliko je broj palindrom ispisati na standardni izlaz 1, ukoliko nije tada ispisati 0, a u slučaju neispravnog ulaza ispisati -1, neispravnim ulazom smatraju se negativni brojevi i brojevi sa brojem cifara manjim ili većim od 4.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
1234 5678	1342 2431	1234 4321	-1234 1234
0	1	1	-1

## 11.2 Programiranje 1, Završni ispit, januar, 23.01.2016.

### 11.2.1 Grupa I

- (4 poena) Napisati C program koji sa standardnog ulaza uitava pozitivan ceo broj  $n$  i na standardni izlaz ispisuje  $n$ -ti lan niza:

$$a_n = \begin{cases} 1 & n = 1 \\ 3 & n = 2 \\ 2a_{n-1} + 3a_{n-2} + 4 & n \geq 3 \end{cases}$$

Neispravnim ulazom se smatra broj manji ili jednak nuli i u tom sluaju na standardni izlaz ispisati -1. Dozvoljeno je korišćenje nizova. Maksimalna vrednost za  $n$  je **2000**.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
-123	1	4	10
-1	1	39	29523

- (7 poena) Napisati funkciju

```
void f3(char s[], char* c, int* br)
```

koja proverava koji karakter se najvie puta pojavio u niski  $s$ . Taj karakter smeta u promenljivu  $c$ , a broj pojavljivanja karaktera u promenljivu  $br$ . Sa standardnog ulaza unosi se linija teksta (moe sadržati beline). Testirati rad funkcije  $f3$  programom koji sa standardnog ulaza uitava nisku i na standarni izlaz ispisati koji karakter se najvie puta pojavio u okviru nje, kao i broj pojavljivanja datog karaktera. Ukoliko postoji vie karaktera iji broj pojavljivanja odgovara maksimalnom broju, ispisati onaj sa najmanjim kodom u ASCII tabeli. Pretpostaviti da se na sistemu koristi ASCII tabela.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
abrakadabra	cvrcak	jorgovan99	s@rm@ ponek@d v@zno
a 5	c 2	9 2	@ 4

- (7 poena) Igra "Minesweeper" sastoji se od pravougaone table izdeljene na polja koja mogu biti bezbedna ili su na njima rasporedjene mine. Sa standardnog ulaza uitavaju se brojevi  $n$  i  $m$  koji oznaavaju dimenzije table. Nako toga unosi se broj  $k$  kojim se navodi koliko mina se nalazi na tabli i  $k$  pozicija  $(i, j)$  koja oznaavaju pozicije na tabli na kojima se nalaze mine ( $i$ -ti red,  $j$ -ta kolona). Korisnik zatim unosi koordinate  $l$  i  $m$  za koje se na standardni izlaz ispisuje broj koliko se mina nalazi na poljima susednim tom polju. Proveravaju se susedna polja u svih 8 pravaca. Ukoliko je polje koje se proverava ba mina ispisati na standardni izlaz **MINA**. Maksimalna dimenzija table je  $100 \times 100$ . Ukoliko je neka od koordinata izvan dimenzija table ili su dimenzije table izvan dozvoljenih granica na standardni izlaz ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:    Izlaz:	Ulaz:    Izlaz:	Ulaz    Izlaz:	Ulaz:    Izlaz:
4 4      2	4 4      MINA	2 3      -1	101 10   -1
3	2	1	1
0 1	0 1	-1 0	45 67
1 2	1 2	2 2	30 31
2 3	2 3		
2 2	2 3		

4. (7 poena) Sluša gradskog prevoza eli da u svakom trenutku ima evidenciju o opterećenju svojih linija. Na linijama saobraćaju autobusi, trolejbusi i tramvaji. Maksimalni kapacitet autobusa je 25, trolejbusa 20 a tramvaja 30 putnika. Broj linije je pozitivan ceo broj manji od 1000.

- a) (1 poen) Definisati strukturu kojim se opisuje vozilo. Svako vozilo zadato je svojim tipom (autobus, trolejbus, tramvaj), linijom na kojoj saobraća i brojem putnika koji se u vozilu nalaze.
- b) (6 poena) Sa standardnog ulaza se unosi broj  $n$  ( $0 \leq n \leq 1000$ ), **n vozila i broj linije**. Za zadati broj linije na standardni izlaz ispisati ukupan broj slobodnih mesta na toj liniji. Koristiti strukturu definisanu pod a). Neispravnim ulazom smatraju se negativan broj putnika, broj putnika veći od dozvoljenog kapaciteta za navedeni tip vozila, tip vozila sa nazivom različitim od navedena tri ili negativan broj linije. U tim slučajevima na standardni izlaz ispisati -1.

Primer 1:	Primer 2:	Primer 3:
Ulaz:                    Izlaz:	Ulaz:                    Izlaz:	Ulaz                    Izlaz:
4                            8	3                            -1	3                            0
autobus    27 18	Autobus 65 23	tramvaj 7    29
trolejbus 28 15	Kombi    1    10	tramvaj 3    15
tramvaj    7    29	minibus 6    21	tramvaj 12 12
autobus    27 24	6	14
27		

---

Primer 4:	Primer 5:
Ulaz:                    Izlaz:	Ulaz:                    Izlaz:
2                            -1	500                       -1
autobus 26 20	
tramvaj 9    32	

## 11.2.2 Grupa II

1. (4 poena) Napisati C program koji za uneti niz celobrojnog tipa i neparne duine  $n$  ispisuje po  $k$  elemenata levo i desno od sredine niza (ne uključujući sredinu). Prvo se unosi  $n$ , zatim niz od  $n$  elemenata, a na kraju i  $k$ .

Neispravnim ulazom se smatra niz parne ili negativne duine, kao i  $k$  koje je negativno ili veći od polovine duine niza. U slučaju neispravnog ulaza ispisati -1 na standardni izlaz.

Smatrati da je maksimalna veličina niza 100 elemenata.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
Ulaz:	Ulaz:	Ulaz:	Ulaz:	Ulaz:
5	9	6	3	5
1 2 3 4 5	9 8 7 6 5 4 3 2 1	1 2 3 4 5 6	1 2 3	10 9 8 7 6
2	1	5	10	-6
Izlaz:	Izlaz:	Izlaz:	Izlaz:	Izlaz:
1 2 3 4	6 4	-1	-1	-1

2. (7 poena) Barkod kodira broj proizvoda dodajui mu kontrolnu cifru. Kontrolna cifra izraunava se kao poslednja cifra zbira jedinica u zapisu svake cifre broja proizvoda. Npr. broj 86012 kodira se kao 1000 0110 0000 0001 0010 a kontrolna cifra je  $(1 + 1 + 1 + 1 + 1) \bmod 10 = 5$ .

Napisati funkciju

```
void kontrolna(char broj_proizvoda[], int *kont)
```

koja izraunava kontrolnu cifru broja proizvoda, koji se zadaje kao niska, i smeta ga u promenljivu kont. Niska moe sadrati beline i druge karaktere, ali ih pri izraunavanju kontrolne cifre treba ignorisati, samo cifre uzeti u obzir.

Napisati program koji sa standardnog ulaza uitava liniju teksta kojom je predstavljen broj proizvoda i testira funkciju kontrolna. Na standardni izlaz ispisati izraunatu kontrolnu cifru. Maksimalna duina niske je 100 karaktera.

Na sistemu se koristi ASCII tabela. Ukoliko ne postoji ni jedna cifra u barkodu, onda je kontrolna cifra 0.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
86012	001-223-4	555 555-555	AB-- 123 --BA
Izlaz:	Izlaz:	Izlaz:	Izlaz:
5	6	8	4

OKRENITE STRANU !!!

3. (7 poena) Napisati program koji ispisuje prosek zbirova svih kolona matrice iji su elementi tipa `double`. Prvo se unosi broj redova matrice  $n$ , zatim broj kolona matrice  $m$ , i onda  $n$  redova sa po  $m$  elemenata. Maksimalna veličina matrice je  $100 \times 100$ . Ukoliko je ulaz neispravan (za vrednosti  $m$  i  $n$ ) prekinuti rad programa i ispisati `-1`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
4 4	3 2	2 4	3 3
0.2 0.4 0.7 1.3	1.23 4.56	0.1 0.2 0.3 0.4	1 0 0
1.5 1.7 2.2 2.5	0 1	10.98 7.65 4.32 1	0 1 0
6.3 -1.2 4.4 5.6	7.89 1	Izlaz:	0 0 1
1.6 2.3 2.8 3.5	Izlaz:	6.2375	Izlaz:
Izlaz:	7.8400		1.000
8.9500			

4. (7 poena) Profesor na jednom predmetu je uveo pravilo da njegov predmet položio svako ko na ispitu osvoji broj poena koji je veći ili jednak od proseka poena umanjeno za 10.

- a) (1 poen) Definisati strukturu kojom se opisuje svaki student sa indeksom (indeks-u-obliku-alas-naloga) i brojem poena koji je osvojio (ceo broj od 0 do 100).
- b) (6 poena) Na ulazu ete dobiti  $n$  ( $0 \leq n \leq 300$ ), broj studenata koji su polagali predmet, i onda  $n$  redova oblika

indeks-u-obliku-alas-naloga broj-poena-na-ispitu

Ispisati na standardni izlaz indekse svih studenata koji su položili ovaj predmet. Koristiti strukturu definisanu pod a).

Smatrati da je indeks pravilno zapisan. U slučaju loše vrednosti za  $n$  ili loše vrednosti za broj poena ispisati `-1`.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
Ulaz:	Ulaz:	Ulaz:	Ulaz:	Ulaz:
4	4	6	4	3
mi12123 80	mr12345 91	mi00001 20	mi11110 100	mi05900 98
mi15512 70	ml54321 80	mi00002 32	mi11111 99	mi13034 120
mi15555 99	mv36925 29	mi00003 96	mi11112 98	mi11234 34
mi13333 40	mi14725 55	mi00004 52	mi11113 87	Izlaz:
Izlaz:	Izlaz:	mi00005 41	Izlaz:	-1
mi12123	mr12345	mi00006 15	mi11110	
mi15512	ml54321	Izlaz:	mi11111	
mi15555	mi14725	mi00003	mi11112	
		mi00004	mi11113	
		mi00005		

### 11.3 Programiranje 1, Završni ispit, februar, 12.02.2016.

1. *URM zadatak* Napisati URM program koji izračunava funkciju:

$$f(x) = \begin{cases} 2(x - y) & x \geq y \\ 0 & \text{inače} \end{cases}$$

2. Sa standardnog ulaza se unose celi, nenegativni brojevi sve dok se ne unese nula. Na standardni izlaz ispisati kvadrat razlike najveceg i najmanjeg od unetih brojeva. U sluaju neispravnog ulaza ispisati -1. Ulaz se smatra neispravnim ukoliko je unet negativan broj ili ukoliko nije unet ni jedan broj osim nule.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
1 2 3 4 5 0	1 2 3 -4 5 0	1 1 1 1 0	0
16	-1	0	-1

3. a) Napisati funkciju

```
void mutacije(char s1[], char s2[], int *br)
```

koja za navedene niske **s1** i **s2** iste duine proverava na koliko mesta se karakteri niski razlikuju i rezultat upisuje u promenljivu **br**. Pri poređenju ignorisati beline.

b) Napisati program koji sa standardnog ulaza uitava dve DNK sekvence (niske karaktera A, T, C ili G) iste duine i testira funkciju **mutacije** ispisujui vrednost promenljive **br** na standardni izlaz. Maksimalna dužina niski je 100 karaktera. U sluaju neispravnog ulaza ispisati -1. Ulaz se smatra neispravnim ukoliko neka od niski sadri karakter koji ne pripada skupu {A, T, C, G} i nije belina ili je jedna niska dua od druge.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
AGTC CGCT AGT	ATCG ATCG ATCG	AGTTGTTGT ATGX	AGGGATGGATGAG
AGTCC GC TAGT	ATCG ATGC ATCG	TTGTATGGA GGAT	TTGATGACGT
Izlaz:	Izlaz:	Izlaz:	Izlaz:
0	1	-1	-1

4. Krtice su organizovano napale batu argarepa. Farmer je napravio pravougaonu mapu bate dimenzija  $n \times m$ , gde je znakom **X** oznaio polje na kome se nalazi krtinjak, dok je netaknuta polja oznaio znakom **-**. Kako je bata velika, farmer eli da bez mnogo muke izrauna broj krtinjaka u proizvoljnom pravougaonom delu svoje bate. Sa standardnog ulaza unose se dimenzije mape **n** i **m**, zatim mapa bate sa oznakama krtinjaka i netaknutih polja. Nakon toga farmer zadaje koordinate (**i1**, **j1**) i (**i2**, **j2**) koje oznaavaju gornji levi i donji desni ugao pravouganika za koji farmer pita koliko krtinjaka je obuhvaeno na mapi tim pravouganikom. Na standardni izlaz ispisati broj krtinjaka u zadatom pravouganiku. Maksimalna dimenzija mape je 100 x 100. U sluaju neispravnih koordinata uglova pravouganika, neispravnih dimenzija mape ili oznaka na tabli van skupa { X, - } na standardni izlaz ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
4 4	4 4	4 4	4 4
- - X -	- - X K	- X - X	- X - X
X - - -	- X - -	X - X -	X - X -
- X - X	X - - -	- X - X	- X - X
X - X -	X X - X	X - X -	X - X -
0 1	1 2	3 4	0 0
2 2	3 4	1 2	3 3



Izlaz:	Izlaz:	Izlaz:	Izlaz:
2	-1	-1	8

5. Vlasnik pekare eli da utvrdi koliko je isplativa prodaja njegovog najskupljeg peciva.

a) Definisati strukturu **Pecivo** koja sadri podatke o imenu peciva (najvie 50 karaktera) i ceni peciva (realan broj tipa double).

b) Sa standardnog ulaza se unosi broj **n** a zatim meseni obraun sa n prodatih komada peciva, pri čemu je naziv peciva u jednom redu a cena u narednom. Na standardni izlaz ispisati ukupnu zaradu od prodaje najskupljeg peciva zaokruenu na dva decimalna mesta. U sluaju negativne cene peciva ili u slučaju da je n manje ili jednako nuli ispisati -1. Pretpostaviti da e samo jedna vrsta peciva imati maksimalnu cenu.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
5	3	-1	5
burek sa mesom	mafin		kroasan sa dzemom
100.50	-50.03		49.99
buhtla sa cokoladom	krofna		kroasan sa dzemom
50.00	56.00		49.99
burek sa mesom	krofna		kroasan sa dzemom
100.50	56.00		49.99
rol virsla			kroasan sa dzemom
75.00			49.99
kroasan sa kremom			kroasan sa dzemom
60.00			49.99
Izlaz:	Izlaz	Izlaz:	Izlaz:
201.00	-1	-1	249.95

## 11.4 1. Grupa, I smer, Programiranje 1 2015/2016, ispit, jun

1. Napisati program koji sa standardnog ulaza učitava prvo pozitivan ceo broj  $n$  ( $0 < n \leq 99$ ), a zatim i  $n$  celih brojeva i izračunava zbir parnih. Izračunati zbir ispisati na standardni izlaz. U slučaju greške (za  $n \leq 0$  ili  $n \geq 100$ ) na standardni izlaz ispisati -1.

Ulaz	5 1 2 3 4 5	5 -1 -2 -3 -4 -5	3 10 -10 10	-3 1 2 3
Izlaz	6	-6	10	-1

2. Napisati funkciju `void f(char s[], char c, int *prva, int* poslednja)` koja u datoj nisci  $s$  pronalizi indekse prvog i poslednjeg pojavljivanja datog karaktera  $c$  i dobijene vrednosti redom smešta u promenljive  $prva$  i  $poslednja$ . Ukoliko se karakter ne pojavljuje u nisci, obe vrednosti postaviti na -1.

Potom napisati program koji sa standardnog ulaza učitava karaktersku nisku (dužine ne veće od 150 karaktera) i jedan karakter i nakon toga poziva funkciju  $f$ , a potom na standarni izlaz ispisuje indekse prvog i poslednjeg pojavljivanja datog karaktera u datoj nisci. Pretpostaviti da je ulaz u ispravnom formatu.

Ulaz	ucionica i	ucionica u	ucionica o	ucionica p
Izlaz	2 5	0 0	3 3	-1 -1

3. Sa standardnog ulaza se zadaje dimenzija kvadratne matrice  $n$  ( $0 < n \leq 99$ ), a zatim elementi matrice koji su celi brojevi. Na standardni izlaz ispisati redni broj vrste koja ima najveći zbir elemenata. U slučaju greške (za  $n \leq 0$  ili  $n \geq 100$ ) na standardni izlaz ispisati -1.

Ulaz	3	5	5	-3
	1 2 3	1 0 0 0 0	1 0 -1 0 -1	1 2 3
	4 5 6	1 1 0 0 0	-1 0 -1 5 0	4 5 6
	7 8 9	1 1 1 0 0	1 -1 -1 0 1	7 8 9
		1 1 1 1 0	1 0 -3 0 -1	
		1 1 1 1 1	0 -1 0 -1 0	
Izlaz	2	4	1	-1

4. Definisati strukturu *Tacka* za predstavljanje tačaka u ravni sa koordinatama tipa *double*. Sa standardnog ulaza se učitava broj  $n$  ( $1 < n \leq 99$ ), zatim niz od  $n$  tačaka tako što se unosi prvo  $x$ , pa  $y$  koordinata za svaku tačku. Za zadate tačke ispisati na standardni izlaz dužinu najduže duži koja se može obrazovati od neke dve tačke iz učitano g niza. Rezultat ispisati na dve decimale. Dužina duži između tačaka  $a(x_1; y_1)$  i  $b(x_2; y_2)$  se računa po formuli

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

U slučaju greške (za  $n \leq 1$  ili  $n \geq 100$ ) na standardni izlaz ispisati -1.

Ulaz	2	3	4	1
	3 4	0.5 0.3	1.2 1.2	0 0
	0 0	-5 3	1.2 1.2	
		3 4	1.2 1.2	
			1.2 1.2	
Izlaz	5.00	8.06	0.00	-1

## 11.5 1. Grupa, I smer, Programiranje 1 2015/2016, ispit, septembar

1. Napisati program koji sa standardnog ulaza učitava dva pozitivna cela broja,  $n$  i  $k$ . Odrediti broj koji se dobije kada se broj  $n$  rotira u levo za  $k$  mesta. U slučaju neispravnog ulaza, na standardni izlaz ispisati -1.

Ulaz	1234 2	101 1	101 2	111 10	-2 3
Izlaz	3412	11	110	111	-1

2. Sa standardnog ulaza se unosi vreme poletanja aviona (u obliku sat:minut:sekunda) i vreme sletanja aviona (let se obavlja u toku 24 sata, ali može biti u dva dana). Odrediti dužinu trajanja leta (u obliku sati:minuti:sekunde).

Ulaz	12:00:00 13:00:00	10:12:23 09:13:24	17:55:42 01:11:32	23:59:59 01:00:01
Izlaz	01:00:00	23:01:01	07:15:50	01:00:02

3. Napisati program koji sa standardnog ulaza učitava neoznačen ceo broj  $n$  ( $1 \leq n \leq 100$ ) i kvadratnu matricu celih brojeva dimenzije  $n \times n$ . Odrediti one elemente matrice koji su najveći u svojoj koloni i najveći u svojoj vrsti (ne ispisivati duplikate, videti primer 2 u tabeli).

Ulaz	3	5	5	3	-3
	1 2 3	1 0 0 0 0	1 0 -1 0 -1	1 2 3	
	4 5 6	1 1 0 0 0	-1 0 -1 5 0	3 2 1	
	7 8 9	1 1 1 0 0	1 -1 -1 0 1	1 2 3	
		1 1 1 1 0	1 0 -3 0 -1		
		1 1 1 1 1	0 -1 0 -1 0		
Izlaz	9	1	1 5	3	-1

4. Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj  $n$  ( $1 \leq n \leq 100$ ), a zatim podatke o instrukcijama koji su zapisani u  $n$  narednih redova. Svaka instrukcija se sastoji iz imena instrukcije (add ili sub) i dva indeksa registara koje instrukcija koristi. Instrukcija add sabira vrednosti prvog navedenog i drugog navedenog registra i rezultat smeta u prvi navedeni registar, dok instrukcija sub izraunava razliku vrednosti prvog i drugog navedenog registra i rezultat smeta u prvi navedeni registar. Registri su indeksirani brojevima od 0 do 9 i inicijalna vrednost svakog registra je jednaka 1. Svaka instrukcija je zapisana u jednom redu programa. Nakon učitanih podataka, sa standardnog ulaza se unosi broj registra i na standardni izlaz se ispisuje tekua vrednost tog registra. U sluaju greke, na standardni izlaz ispisati -1.

Ulaz	3 add 3 4 add 3 5 add 3 3 3	2 add 0 1 sub 0 0 0	4 add 0 1 add 1 2 sub 4 6 sub 4 2 7	-1 add 4 5 2
Izlaz	6	0	1	-1

# Glava 12

## Akademaska godina 2016/2017

### 12.1 Programiranje 1 - predlog kolokvijuma

#### 12.1.1 Grupa 1

1. Napisati URM program koji izraunava funkciju:

$$f(x, y, z) = z + 10 * y$$

Nacrtati odgovarajui dijagram.

2. Napisati program koji za uneti karakter (< ili >) i dva cela pozitivna trocifrena broja, poredi brojeve prema datom karakteru i ako je uslov ispunjen zamenjuje najveću cifru prvog broja sa najmanjom cifrom drugog broja, a ako uslov nije ispunjen zamenjuje najmanju cifru prvog broja sa najvećom cifrom drugog broja. Ukoliko ima više najmanjih/najvećih cifara u okviru nekog broja, uzeti onu sa najmanjom težinom. Na standardni izlaz ispisati brojeve nakon zamene cifara. U slucaju greke, na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:
Ulaz: < 123 456	Ulaz: > 123 456	Ulaz: < 444 555	Ulaz: + 123 123	Ulaz: < 1234 123
Izlaz: 124 356	Izlaz: 623 451	Izlaz: 445 554	Izlaz: -1	Izlaz: -1

3. Napisati program koji za unetu registraciju, formata GGBBBB, gde je GG skraenica za grad i moe imati vrednosti BG - Beograd, NS - Novi Sad, NI - Nis, SU - Subotica ili VA - Valjevo, a BBBB redni broj, ispisuje puno ime grada i redni broj bez vodeih nula. U slucaju greke, na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:
Ulaz: BG1234	Ulaz: BG0789	Ulaz: SU12CD	Ulaz: NS0001	Ulaz: AB1234
Izlaz: Beograd 1234	Izlaz: Beograd 789	Izlaz: -1	Izlaz: Novi Sad 1	Izlaz: -1

#### 12.1.2 Grupa 2

1. Napisati URM program koji izraunava funkciju:

$$f(x, y, z) = \begin{cases} 3 * x + y - z & \text{ako je } y > z \\ 0 & \text{ako je } y \leq z \end{cases}$$

Nacrtati odgovarajui dijagram.

- Napisati program koji za uneti pozitivan etvorocifreni broj, ako je uneti broj broj palindrom, ispisuje broj koji se dobija izbacivanjem cifara sa parnih pozicija iz unetog broja, ili, ako uneti broj nije palindrom, ispisuje broj koji se dobija izbacivanjem cifara sa neparnih pozicija iz unetog broja. Pozicije se broje sa desna na levo i počinju od 0. U slučaju greke, na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:
Ulaz: 1221	Ulaz: 7531	Ulaz: 4444	Ulaz: 455	Ulaz: -4567
Izlaz: 12	Izlaz: 51	Izlaz: 44	Izlaz: -1	Izlaz: -1

- Napisati program za ifrovanje koji uitava karakter  $c$  i pozitivan broj  $b$  ifruje uneti karakter na sledeci nain:

- Ako je uneti karakter malo ili veliko slovo, njegova ifra se dobija ciklinim pomeranjem u desno za  $b$  mesta (npr. za karakter  $y$  i broj 5 dobija se slovo  $d$ . Slinio za karakter  $Y$  i broj 5 dobija se slovo  $D$ ).
- Ako je uneti karakter cifra, njegova ifra je broj koji se dobija kada se unetoj cifri doda njen (celobrojni) koren (npr. za unetu 8, sifra je  $8 + \sqrt{8} = 10$ ).
- Ako je uneti karakter zagrada (neka od: (, ), {, }, [ ili ]), karakter ostaje nepromenjen.
- U suprotnom, ifra je \*.

Na standardni izlaz ispisati ifru unetog karaktera. U sluaju greke na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:	Primer6:
Ulaz: a 3	Ulaz: 5 1	Ulaz: Z 54	Ulaz: + 3	Ulaz: h -5	Ulaz: ( 5
Izlaz: d	Izlaz: 7	Izlaz: B	Izlaz: *	Izlaz: -1	Izlaz: (

### 12.1.3 Grupa 3

- Napisati URM program koji izraunava funkciju:

$$f(x, y, z) = \begin{cases} x * y + 4 * z & \text{ako je } x \leq y \\ 0 & \text{ako je } x > y \end{cases}$$

Nacrtati odgovarajui dijagram.

- Napisati program koji za dva uneta pozitivna dvocifrena broja, ako je prvi broj vei, ispisuje broj koji se dobija tako sto se dva uneta broja spoje, pa im se zatim izbace sve cifre ija je vrednost paran broj, a u suprotnom ispisuje broj koji se dobija spajanjem dva uneta broja, a zatim izbacivanjem svih cifara ija je vrednost neparan broj. U slucaju greke, na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:
Ulaz: 14 22	Ulaz: -12 13	Ulaz: 1234 5	Ulaz: 33 11	Ulaz: 11 33
Izlaz: 422	Izlaz: -1	Izlaz: -1	Izlaz: 3311	Izlaz: 0

3. Sa standardnog ulaza se učitavaju pozitivan realan broj  $n$  i karakter  $k$  koji predstavlja mernu jedinicu za vreme (s - sekund, d - minut, h - sat). Na standardni izlaz ispisati kojom brzinom se kree objekat ako za zadatu vrednost vremena prelazi put od 10m. Rezultat izraziti u m/s i ispisati ga na etiri decimalne. Brzina = predjeni put / vreme. U slucaju greke, na standardni izlaz ispisati -1.

Primer1:	Primer2:	Primer3:	Primer4:	Primer5:
Ulaz: 23.4 s	Ulaz: 20 d	Ulaz: 0.01 h	Ulaz: -3 s	Ulaz: 3.5 t
Izlaz: 0.4274	Izlaz: 0.0083	Izlaz: 0.2778	Izlaz: -1	Izlaz: -1

## 12.2 Programiranje 1, Završni ispit, januar, 24.01.2017.

### 12.2.1 Grupa I

1. (5 poena) Program učitava ceo broj. Odrediti broj koji se dobija izbacivanjem parnih cifara iz polaznog broja. Pri tome, ne menjati znak broja. Potrebno je zaista kreirati novi broj, a ne samo ispisati cifre u nekom poretku.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
-896754	86	897612	7753	-90642
-975	0	971	7753	-9

2. (5 poena) Program učitava rei sve do kraja ulaza (EOF-a). Ispisati sve one rei u tekstu koje se mogu sastaviti od karaktera rei "programiranje" (pri tome ne moraju biti upotrebljeni svi karakteri). Zanimariti veličinu slova. Maksimalna dužina rei je 20 karaktera. (Napomena: Rei se ne moraju pamtit u nizu, već se mogu odmah ispisivati nakon nailaska na reč koja zadovoljava traženi kriterijum.)

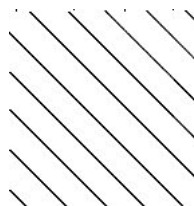
Primer 1:	Primer 2:	Primer 3:
Danas grami tako	Programi su zanimljivi.	Programiranje je zabavno.
grami	Programi	je

---

Primer 4:  
Mir je gram

Mir je gram

3. (7 poena) Unosi se ceo pozitivan broj  $n$ , a potom i celobrojna kvadratna matrica dimenzije  $n \times n$ . Potom se unosi ceo pozitivan broj  $k$  ( $0 < k \leq 2 * n - 1$ ). Ispisati  $k$ -tu dijagonalu matrice. Dijagonalama su pridrueni redni brojevi, počevši od 1 od donjeg levog ugla, pa do gornjeg desnog ugla (kao dati primer na slici). Maksimalna dimenzija matrice je  $50 \times 50$ . U slucaju greške pri unosu podataka ispisati -1.



Primer 1:                      Primer 2:                      Primer 3:                      Primer 4:

5	5	2	55
1 2 3 4 5	1 2 3 4 5	1 1	
-23 8 11 -9 15	-23 8 11 -9 15	2 2	
6 7 8 23 11	6 7 8 23 11	4	
14 0 -1 -2 8	14 0 -1 -2 8		
5 4 10 30 40	5 4 10 30 40		
4	7		
-23 7 -1 30	3 -9 11	-1	-1

4. (8 poena) Pera igra igru UNO.

- (1 poen) Definirati strukturu kojom se opisuju karte. Svaka karta je označena bojom (tref, karo, pik, herc) i brojem (1-10, 12 (pub), 13 (dama), 14 (kralj)).
- (7 poena) Program unosi broj karata koje ima Pera, a onda i karte. Program potom učitava kartu koja se nalazi na stolu. Ispisati sve karte koje može da igra Pera. Pera može da igra onu kartu koja se poklapa po boji ili po broju sa kartom na stolu. Pera može da ima najviše 51 kartu.

U slučaju greške prilikom unosa podataka ispisati -1 i izaći iz programa. Unos podataka se može prekinuti čim se pokuša unos neispravnog podatka.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
4	2	4	5
herc 10	tt 2	herc 0	herc 1
tref 12	tref 1	tref 11	karo 2
karo 8		karo 23	herc 7
pik 8		pik -12	tref 8
			tref 14
tref 8			herc 8
tref 12	-1	-1	herc 1
karo 8		(napomena: u ovom test	herc 7
pik 8		primeru ni jedan	tref 8
		ulaz nije ispravan)	

### 12.2.2 Grupa 2

5. (5 poena) Program učitava ceo broj. Odrediti broj koji se dobija izbacivanjem svih cifara koje su vee od aritmetike sredine svih cifara u zapisu unetog broja. Pri tome, ne menjati znak broja. Potrebno je zaista kreirati novi broj, a ne samo ispisati cifre u nekom poretku.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
-896754	86	897612	7753	-90642
-654	6	12	3	-2

6. (5 poena) Napisati C funkciju `int orf(char rnk[])` koja za zadatu nisku `rnk`, maksimalne duzine do 100 karaktera koja se sastoji samo od velikih slova A, U, C, G, ispisuje koliko postoji moguih poetaka translacije u protein, tanije koliko puta se pojavljuje podsekvencu AUG. U sluaju unosa neispravne niske, funkcija treba da vrati -1. Napisati program koji uitava rei sve do kraja ulaza (EOF-a) i za svaku re ispisuje rezultat izvravanje funkcije `orf`.

Primer 1:  
AUGA AUGAUG

Primer 2:  
GGAUG GACC GGTT

Primer 3:  
CAGCAG AUCAUG GSAD

Primer 4:  
[prazan ulaz]

1 2

1 0 -1

0 1 -1

[prazan izlaz]

7. (7 poena) Napisati C program koji za unetu pozitivnu celobrojnu vrednost  $n$  i matricu dimenzija  $n \times n$  uitanu sa standardnog ulaza, na standardni izlaz ispisuje elemente sa glavne i sporedne dijagonale matrice koji su manji od aritmetike sredine svih elemenata u matrici. Maksimalna dimenzija matrice je 50. Ukoliko nema elemenata koji zadovoljavaju traeni kriterijum, ispisati "Nema".

Primer 1:

5  
1 2 3 4 5  
-23 8 11 -9 15  
6 7 8 23 11  
14 0 -1 -2 8  
5 4 10 30 40

1 -2 5 -9 0 5

Primer 2:

3  
1 2 3  
-23 8 11  
6 7 8

1

Primer 3:

2  
1 1  
2 2

1 1

Primer 4:

55

-1

Primer 5:

3  
0 0 0  
0 0 0  
0 0 0

Nema



8. (8 poena)

- a) (1 poen) Definirati strukturu ELEMENT koja uva naziv hemijskog elementa duine najvie dva karaktera, celobrojni indikator da li je element metal ili ne i masu elementa u gramima koja je realan broj.
- a) (7 poen) Napisati program koji sa stanadrnog ulaza uitava pozitivan ceo broj  $n$  a zatim podatke o  $n$  elemenata koji su nadjeni u uzorku od 100g - naziv elementa, indikator da li je element metal ili ne (0 ako element nije metal ili 1 ako jeste) i masu elementa u gramima. Potrebno je da suma masa svih  $n$  elemenata iznosi najmanje 99g a najvie 101g. Na standardni izlaz ispisati procenat metala i procenat nemetala u uzorku zaokruen na dve decimale. U sluaju da gramaa uzorka nije u zadatim granicama ili da vrednost indikatora nije ispravna, na standardni izlaz ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:	Primer 5:
3	1	2	2	4
Na 1 40.1	Na 1 100	Na 1 67.87	Fe 1 43	S 0 23.11
Al 1 19.9		Al 1 65.34	Mg 1 33	Cl 0 12.23
Cl 0 39				K 1 13.34
				He 0 52.56
39.00 60.00	0.00 100.00	-1	-1	87.90 12.33

### 12.3 Programiranje 1, Završni ispit, februar, 02.02.2017.

1. Napisati program kojim se sa standardnog ulaza unose celi brojevi do unosa broja 0 i ispisuju oni brojevi koji imaju neparan broj cifara. Broj brojeva nije unapred poznat.

Primer 1	Primer 2	Primer 3	Primer 4
Ulaz:	Ulaz:	Ulaz:	Ulaz:
1 10 100 1000 0	11 222 33 0	42 -3 0	0
Izlaz:	Izlaz:	Izlaz:	Izlaz:
1 100	222	-3	

2. Napisati program koji sa standardnog ulaza uitava nisku  $s$  i ceo broj  $n$ ,  $1 \leq n \leq 100$ , a zatim  $n$  niski i na standardni izlaz ispisuje sve niske za koje vai da im je  $s$  prefiks ili sufiks. Maksimalna duina svih niski koje se uitavaju je 100 karaktera. U sluaju neispravnog ulaza na standardni izlaz ispisati -1.

Primer 1	Primer 2	Primer 3	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
aba	da	ana	rad
101	3	1	3
	abeceda	banane	BEOGRAD
	dabar		radnicki
	radan		parada
Izlaz:	Izlaz:	Izlaz:	Izlaz:
-1	abeceda		radnicki
	dabar		

3. Napisati program koji sa standardnog ulaza učitava ceo broj  $n$ ,  $1 \leq n \leq 50$ , cele brojeve  $k$  i  $l$ ,  $0 \leq k, l < n$  a zatim kvadratnu matricu dimenzija  $n \times n$  iji su elementi celi brojevi. Na standardni izlaz ispisati sumu elemenata koji se nalaze na ivicama podmatrice unete matrice iji gornji levi ugao ima koordinate  $(k,k)$  a donji desni  $(l,l)$ . U sluaju neispravnog ulaza, na standardni izlaz ispisati -1

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
2	3	3	4
0 1	0 2	1 2	3 4
1 2	1 2 3	1 2 3	
3 4	4 5 6	4 5 6	
	7 8 9	7 8 9	
Izlaz:	Izlaz:	Izlaz:	Izlaz:
10	40	28	-1
Objanjenje:	Objanjenje:	Objanjenje:	
1 + 2 = 10	1 + 2 + 3 = 40	5 + 6 = 28	
+3 + 4	+4 + 6	+8 + 9	
	+7 + 8 + 9		

4. a) Definisati strukturu *DUEL* koja sadri podatke o nazivima dva fudbalska kluba (niske duine najvie 20 karaktera), broju golova koje je domain postigao, broju golova koje je gost postigao kao i nazivu stadiona (niska duine najvie 30 karaktera) na kome je duel odigran.

b) Sa standardnog ulaza se unosi broj  $n$ ,  $1 \leq n \leq 1000$ , a zatim  $n$  duela u obliku naziv kluba domaina, naziv gostujuceg kluba, naziv stadiona, broj golova domaina broj golova gosta. Brojevi golova su nenegativni celi brojevi. Na standardni izlaz ispisati naziv stadiona na kome je postignuta najvea gol razlika (apsolutna vrednost razlike golova domaina i gosta). Nazivi klubova i stadiona se unose u zasebnim linijama teksta, pretpostaviti da nee postojati vie stadiona sa najveom gol razlikom. U sluaju neispravnog ulaza, na standardni izlaz ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
Ulaz:	Ulaz:	Ulaz:	Ulaz:
4	2	1	2
Barcelona	Crvena Zvezda	FK Djerdap	OFK Beograd
Real Madrid	Juventus	Man City	BSK Borca
Camp Nou	Rajko Mitic	Stadion pod Zutim bregom	Omladinski stadion
3 0	0 1	7 0	1 -1
A.C. Milan	Partizan	Izlaz:	Izlaz:
Juventus	Fenerbahce	Stadion pod Zutim bregom	-1
Giuseppe Meazza	Stadion JNA		
1 3	0 0		
FK Javor	Izlaz:		
FK Sevojno	Rajko Mitic		
Stadion kraj Moravice			

12 10

Barcelona  
Man Utd  
Camp Nou  
1 1

Izlaz:  
Camp Nou