

Zadatak 1 Sa standardnog ulaza unosi se neoznačen ceo broj. Napisati program koji određuje i ispisuje sumu cifara tog broja.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
45678	5672	555555444	-29
30	20	42	56

Zadatak 2 Sa standardnog ulaza se unosi jedna linija teksta. Napisati program koji prikazuje koliko puta se javilo svako od slova engleskog alfabeta (ne praviti razliku između velikih i malih slova).

Primer 1:
haHJjKL

a:1 b:0 c:0 d:0 e:0 f:0 g:0 h:2 i:0 j:2 k:1 l:1 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:0 z:0

Primer 2:
DanaS j3 _j_utRo laBU78d

a:3 b:1 c:0 d:2 e:0 f:0 g:0 h:2 i:0 j:2 k:0 l:1 m:0 n:1 o:1 p:0 q:0 r:1 s:1
t:1 u:2 v:0 w:0 x:0 y:0 z:0

Primer 3:
Sao PaoLo 1998 _JuZna Amerika90

a:5 b:0 c:0 d:2 e:1 f:0 g:0 h:0 i:1 j:1 k:1 l:1 m:1 n:1 o:3 p:1 q:0 r:1 s:1
t:0 u:1 v:0 w:0 x:0 y:0 z:0

Primer 4:
Ixxx kk 3yyy 4qqq

a:0 b:0 c:0 d:0 e:0 f:0 g:0 h:0 i:1 j:0 k:2 l:0 m:0 n:0 o:0 p:0 q:4 r:0 s:0
t:0 u:0 v:0 w:0 x:3 y:3 z:0

Zadatak 3 Napisati void funkciju koja za uneti broj sekundi proteklih od ponoći određuje i vraća (ne ispisuje na izlaz) broj sati, minuta i sekundi. Napisati i program koji testira ovu funkciju. U slučaju neispravnog unosa ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
23456	71819	92000	-89
6:30:56	19:56:59	-1	-1

Zadatak 4 2013. godina je prva posle 25 godina (1988-2012.) koja u zapisu ne sadrži dve iste cifre. Sa standardnog ulaza se unose dve četvorocifrene godine g_1 i g_2 . Napisati program koji:

- ispisuje broj godina g takve da je $g_1 \leq g \leq g_2$ i g ima dve iste cifre u zapisu. [2]
- određuje najduži niz uzastopnih godina započet u godini g takvoj da je $g_1 \leq g \leq g_2$ u kome sve godine imaju bar dve iste cifre (ispisuje se prva i poslednja godinu u nizu). U slučaju da takav niz godina ne postoji ispisati -1. [4]

Proveriti ispravnost podataka - da li je godina pozitivan četvorocifren broj i da li je prva godina manja od druge. Ako su podaci neispravni ispisati -1.

Primer 1:	Primer 2:	Primer 3:	Primer 4:
1990 2343	1782 1784	1801 1901	200 2000
a: 226 b: 2199 2300	a: 0 b: -1	a: 45 b: 1810 1819	-1

Zadatak 5 Napisati program koji za uneti neoznačen broj 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 $n=3$

1 2 3
1 3
1

Primer 2:
za $n=1$

1

Primer 3:
za $n=7$

1 2 3 4 5 6 7
1 3 5 7
1 4 7
1 5
1 6
1 7
1

Primer 4:
za $n=-23$

Zadatak 6 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

zRVo vma

Primer 2:
12345AbcD

12345D

Primer 3:
JeD1aN D52Va.

JeD1N D52Va.

Primer 4:
abcd efg

d g

Zadatak 7 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

Zadatak 8 Sa standardnog ulaza učitava se broj n , a zatim i kvadratna matrica koja sadrži brojeve tipa `double` dimenzije $n \times n$. Napisati program koji izračunava i ispisuje 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:
3
2 3.2 4
7 8.8 1
2.3 1 1

-2.10

Primer 2:
4
2.3 1 12 8
4 -8.2 7 14.5
1 -2.5 9 11
3 4.3 -5.7 2

49.4

Primer 3:
GRESKA

Primer 4:
-4

GRESKA

Zadatak 9 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`).

Zadatak 10 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.

Zadatak 11 Odrediti dužinu najdužeg uspona u datom nizu a . Uspon od i do j u nizu a je opisan sa: za vrednosti k_1 i k_2 takve da je $i \leq k_1 < k_2 \leq j$ važi $a[k_1] < a[k_2]$. Njegova dužina je tada $j-i+1$.

Zadatak 12 (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 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`).

Zadatak 13 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.

Zadatak 14 Sa standardnog ulaza učitava se n ($0 < n \leq 200$), a potom i spisak (dužine n) engleskih reči i njihov prevod na srpski jezik. Potom se učitava jedna reč sa standardnog ulaza. Na standardni izlaz ispisati odgovarajući prevod date reči ili podatak o tome da se reč ne nalazi na spisku.

```
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 recniku. U programu se mogu koristiti funkcije iz zaglavlja string.h.

Zadatak 15 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
```

Zadatak 16 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.

Zadatak 17 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 $11 \cdot 12^3 + 11 \cdot 12^2 + 1 \cdot 12^1 + 0 \cdot 12^0 = 20604$.

Zadatak 18 Definisati strukturu koja opisuje automobil atributima marka i cena (marka je karakterska niska dužine ne više od 20 karaktera, a cena je ceo broj). Sa standardnog ulaza na početku se unosi broj automobila, a potom redom podatke o automobilima. Uz pretpostavku da može biti 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).

Zadatak 19 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) Sa standardnog ulaza se unose dva cela broja n ($0 < n \leq 100$) i k . Potom se unosi n reči maksimalne dužine 20. Svaku reč rotirati za k mesta i da tako dobijene reči ispisati na standardni izlaz.

Zadatak 20 Definirati strukturu koja opisuje dete atributima ime deteta (ne veće od 20 karaktera), pol deteta (m ili z) i ocena. Ocenu je svako dete dalo radu obdaništa. Maksimalan broj dece je 100. Napisati program koji:

- a) Sa standardnog ulaza se unosi n , a potom podaci o n dece. Koristiti strukturu:

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

- b) ispisati na standardni izlaz statistiku: koliko ima dečaka, a koliko devojčica i prosečnu ocenu. Potom ispisuje imena dece brojnijeg pola.

Zadatak 21 Sa standardnog ulaza se unosi N ($N \geq 2$). Napisati program koji na standardni izlaz ispisuje sledeću sliku: za $N=2$:

```
*
* *
* *
```

za $N=3$:

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

za $N=4$:

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

Zadatak 22 Napisati funkciju `void sifrat(char* rec, char* kljuc)` koja šifrira `rec` na sledeći način: za svako slovo reči `rec` i odgovarajuće slovo `kljuc` određuje koliki je (alfabetски) razmak između njih i označimo taj broj sa k . Potom to slovo `rec` zamenjuje k -tim slovom alfabeta. Podrazumeva se da je `kljuc` duži od `rec`. PRIMER: za `rec = bac` i `kljuc = dfge` rezultat je `bed`

Zadatak 23 Napisati URM program i C program koji izračunavaju sledeću funkciju:

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

Zadatak 24 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

Zadatak 25 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.

Zadatak 26 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 zadatu 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.

Zadatak 27 a) Uvesti tip podataka `Sifra` kojim se opisuje način šifrovanja alfanumeričkih karaktera. Svaka šifra se opisuje celobrojnomo vrednošću `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 zadatu 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) Sa standardnog ulaza se zadaje način šifrovanja i to u obliku `2 D 5 L` (šifra može biti i duža). Potom se učitava `n` i `n` reči sa standardnog ulaza (maksimalna dužina reči je 20 karaktera). Ispisati reči na standardni izlaz nakon primenjenih svih zadatih načina šifrovanja.

Zadatak 28 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.

Zadatak 29 • Definirati tip podataka `TACKA` pogodan za predstavljanje tačke Dekartovske ravni (čije su `x` i `y` koordinate podaci tipa `double`).

- Definirati funkciju `double rastojanje(TACKA a, TACKA b)` koja izračunava rastojanje između dve tačke.
- Definirati 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.
- Definirati funkciju `double obim(TACKA* poligon, unsigned n)` koja izračunava obim poligona sa `n` tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
- Definirati funkciju `double maksimalna_stranica(TACKA* poligon, unsigned n)` koja izračunava dužinu najduže stranice poligona sa `n` tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
- Definirati funkciju `main` u kojoj se sa standardnog ulaza učitava celobrojna nenegativna vrednost `N` ($0 < N \leq 100$). Inače, poziva se funkcija `ucitaj_poligon`. Ukoliko je uspešno učitano `m` tačaka (`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 `?`).

Zadatak 30 Napisati URM program koji izračunava funkciju

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

Zadatak 31 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 -0.8391

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

Zadatak 32 *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:

Ulaz: 9 0 0

Izlaz: 10800

Primer 2:

Ulaz: 12 0 1

Izlaz: 86399

Primer 3:

Ulaz: 17 24 6

Izlaz: 66954

Primer 4:

Ulaz: 12 7 87

Izlaz: -1

Zadatak 33 *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