

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

Primer 1: 45678	Primer 2: 5672	Primer 3: 555555444	Primer 4: -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 4qqqq

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: 23456	Primer 2: 71819	Primer 3: 92000	Primer 4: -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 se prva i poslednja godinu u nizu). U slučaju da takv 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: 1990 2343	Primer 2: 1782 1784	Primer 3: 1801 1901	Primer 4: 200 2000
a: 226 b: 2199 2300	a: 0 b: -1	a: 45 b: 1810 1819	-1

Zadatak 5 U datoteci čije se ime navodi sa standardnog ulaza 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. iz njega uklanja sve duplikate i u datoteku rez.txt ispisuje transformisani niz

[4]

U slučaju greške ispisati -1.

Primer 1:		Primer 2:
./a.out dat1.txt		./a.out dat2.txt
dat1.txt: 12 jha14 hahaha deda mraz deda		dat2.txt: 14
mraz deda deda jase konj konj konj		so secer supa so ljuto secer kiselo slatko
		ljuto
jha14 hahaha deda mraz deda mraz deda		paprika, ljuta paprika, ljuto dete
deda jase konj konj konj		
		so secer supa so ljuto secer kiselo slatko
rez.txt: jha14 hahaha deda mraz jase konj		ljuto paprika, ljuta paprika, ljuto dete
		rez.txt: so secer supa ljuto kiselo slatko
		paprika, ljuta dete

Primer 3:		Primer 4:
./a.out dat3.txt		./a.out
dat.txt: 17 Buducnost televizije su ultra HD, odnosno 4K		
uredaji koji imaju ogromnu dijagonalu ekrana i znacajno vise piksela		-1
Buducnost televizije su ultra HD, odnosno 4K		
uredaji koji imaju ogromnu dijagonalu ekrana i znacajno vise piksela		
rez.txt: Buducnost televizije su ultra HD, odnosno 4K		
uredaji koji imaju ogromnu dijagonalu ekrana i znacajno vise piksela		

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

Zadatak 7 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 8 U datoteci čije se ime navodi sa standardnog ulaza 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 podulaku.

[4]

U slučaju greške ispisati -1.

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

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

Zadatak 9 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:	Primer 2:	Primer 3:	Primer 4:
5 -3 4 -5 8 2	-2	7 -12 6 7 -2 12 0 15	3 23 50 31
13	-1	27	27

Zadatak 10 Iz datoteke čije se ime navodi 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 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:	Primer 2:	Primer 3:	Primer 4:
./a.out ulaz.txt	./a.out matrica.txt	./a.out	./a.out dat.txt
ulaz.txt:	matrica.txt:	rezultat.txt:	dat.txt:
3	4	GRESKA	-4
2 3.2 4	2.3 1 12 8		
7 8.8 1	4 -8.2 7 14.5		
2.3 1 1	1 -2.5 9 11		rezultat.txt:
rezultat.txt:	3 4.3 -5.7 2		GRESKA
-2.10	rezultat.txt:		
	49.4		

Zadatak 11 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 12 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 13 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[k1] < a[k2]`. Njegova dužina je tada $j-i+1$.

Zadatak 14 Sastaviti program koji prima sa standardnog ulaza ime datoteke koju treba otvoriti. Tu datoteku treba pročitati i ispisati (na standardnom izlazu) koja cifra (međ u svim ciframa koje se pojavljuju u datoteci) ima najveći broj pojavljivanja.

Zadatak 15 (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`).

Zadatak 16 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.

Zadatak 17 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 18 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"
```

Zadatak 19 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 recniku`. U programu se mogu koristiti funkcije iz zaglavlja `string.h`.

Zadatak 20 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`.

Zadatak 21 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 22 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`.

Zadatak 23 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 24 Napisati funkciju koja vrši prevoenje 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 $11 * 12^3 + 11 * 12^2 + 1 * 12^1 + 0 * 12^0 = 20604$.

Zadatak 25 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.

Zadatak 26 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.

Zadatak 27 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;
} DETE;
```

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

Zadatak 28 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 $N=4$:

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

Zadatak 29 Napisati funkciju `void sifrat(char* rec, char* kljuc)` koja šifruje `rec` na sledeći način: za svako slovo reči `rec` i odgovarajuće slovo ključa određuje koliki je (alfabetски) razmak između njih i označimo taj broj sa k . Potom to slovo reči zamenjuje k -tim slovom alfabeta. Podrazumeva se da je ključ duži od reči. PRIMER: za `rec = bac` i `kljuc = dfge` rezultat je `bed`

Zadatak 30 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 31 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 32 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 33 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 34 a) Uvesti tip podataka `Sifra` kojim se opisuje način šifrovanja alfanumeričkih karaktera. Svaka šifra se opisuje celobrojnom 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, \dots , x sa z, y sa a, z sa $b, 1$ sa $3, \dots, 8$ 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 alfanumeričkih karaktera. Šifrovano prepisati sadržaj ove datoteke u datoteku `izlaz.txt`, redom po svim zadatim načinima šifrovanja.

Zadatak 35 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 36 • 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 izračunava rastojanje između 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 izračunava 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 izračunava 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 uspeła 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 `?`).

Zadatak 37 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 38 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 39 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 40 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

Zadatak 41 1. *Definisati strukturu Tacka za predstavljanje tačaka u ravni sa koordinatama tipa double.*

2. *Napisati funkciju Rastojanje koja izračunava rastojanje između dve tačke.*
3. *Napisati funkciju Obim koja izračunava obim poligona zadatog nizom tacaka.*
4. *Napisati funkciju Povrsina koja izračunava površinu poligona zadatog nizom tacaka (koristiti Heronov obrazac: $P = \sqrt{s(s-a)(s-b)(s-c)}$ pri čemu je s poluobim trougla sa stranicama a, b i c).*
5. *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