

ISPIT IZ PREDMETA “UVOD U PROGRAMIRANJE”

ZADACI SU PODIJELJENI U 3 SKUPINE. ZA PROLAZNU OCJENU, STUDENT MORA IMATI PO JEDAN KOREKTAN ZADATAK OZ SVAKE OD SKUPINA. ZA MAKSIMALNU OCJENU, STUDENT TREBA IMATI PO DVA KOREKTNA ZADATKA IZ SVAKE OD SKUPINA. STUDENTI NE MORAJU RADITI ZADATKE IZ ONIH SKUPINA IZ KOJE SU ZADOVOLJILI ODGOVARAJUĆE KOLOKVIJE TOKOM SEMESTRA.

Skupina 1.

1. Napišite program koji traži od korisnika da unese prirodne brojeve M i N ($M < 80$, $N < 20$). Nakon toga, program treba da iscrta pravougaonik sastavljen od slova abecede, pri čemu su M i N redom širina i visina pravougaonika. Zakonitost kako se redaju slova možete vidjeti iz sljedećeg primjera, koji pokazuje izgled pravougaonika za $M=20$ i $N=7$;

```
AAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBB
ABCCCCCCCCCCCCCCCCBA
ABCDDDDDDDDDDDDDCBA
ABCCCCCCCCCCCCCCCCBA
BBBBBBBBBBBBBBBBBBBBB
AAAAAAAAAAAAAAAAAAAAA
```

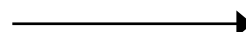
2. Napišite program koji će prvo tražiti od korisnika da unese prirodan broj N, a zatim N realnih brojeva a_1, a_2, \dots, a_N . Nakon toga, program treba da izračuna i ispiše vrijednost izraza

$$a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{\dots + \frac{1}{a_N}}}}$$

3. Par prostih brojeva koji se međusobno razlikuju za 2 nazivaju se *prosti brojevi blizanci*. Na primjer, prosti brojevi blizanci su parovi (2, 3), (17,19), (56,61), itd. Napišite program koji za uneseni prirodni broj n sa tastature ispisuje prvih n parova blizanaca prostih brojeva.

Skupina 2.

1. Napišite funkciju “IzbaciNule” koja ima kao parametar neki prirodan broj, a koja daje kao rezultat isti taj broj, samo iz kojeg su izbačene sve cifre koje su nule. Na primjer, ukoliko se kao parametar funkciji proslijedi broj 35020040, rezultat funkcije treba da bude 3524. Ukoliko broj ne sadrži niti jednu nulu kao svoju cifru, onda je on sam ujedno i rezultat funkcije. Napisanu funkciju trebate demonstrirati u kratkom testnom programu na brojevima koji se unose sa tastature. Program treba da završi rad kada se kao broj unese 0.
2. Za neki element a_n nekog niza brojeva kaže se da je *lokalni maksimum* tog niza brojeva ukoliko je veći od oba svoja susjeda, tj. ukoliko je $a_n > a_{n-1}$ i $a_n > a_{n+1}$. Analogno se definira i *lokalni minimum* niza brojeva. Napišite generičku funkciju koja prima četiri parametra, od kojih je prvi parametar niz elemenata proizvoljnog tipa, dok je drugi parametar broj elemenata u tom nizu. Funkcija treba da pronađe koliko u tom nizu ima lokalnih maksimuma i lokalnih minimuma i da smjesti pronađene brojeve redom u treći i četvrti parametar funkcije. Napisanu funkciju demonstrirajte u kratkom testnom programu u kojem ćete unijeti niz brojeva sa tastature (broj elemenata se također zadaje sa tastature, i neće biti veći od 100), a koji će zatim ispisati koliko u tom nizu brojeva ima lokalnih maksimuma a koliko lokalnih minimuma.



3. U teoriji brojeva, za brojeve m i n kažemo da čine *prijateljski par brojeva* ukoliko je suma svih pravih djelilaca jednog od njih, (tj. svih djelilaca osim njega samog), jednaka onom drugom, i obrnuto. Na primjer, brojevi 220 i 284 čine prijateljski par. Zaista, svi pravi djelci broja 220 su 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 i 110, čija je suma 284, dok svi pravi djelci broja 284 glase 1, 2, 4, 71 i 142, i njihova suma je 220. Napravite funkciju sa dva cjelobrojna parametra koja vraća logičku vrijednost **"true"** ako i samo ako parametri čine prijateljski par brojeva, dok u suprotnom vraća logičku vrijednost **"false"**. Napisanu funkciju trebate demonstrirati u kratkom testnom programu koji za dva broja unesena sa tastature ispituje i ispisuje da li su prijateljski par ili ne. Radi testiranja, navedimo da neki od prijateljskih parova brojeva, pored već navedenog para (220,284) glase (11843 1210), (2620, 2924), (5020, 5564), itd.

Skupina 3.

1. Napišite rekurzivnu funkciju sa jednim cjelobrojnim parametrom, koja vraća kao rezultat sumu cifara broja koji je proslijeđen kao parametar. Pri tome se rad funkcije mora isključivo zasnivati na rekurziji, odnosno sama funkcija ne smije sadržavati nikakav tip petlji (**"for"**, **"while"** ili **"do" – "while"**). Napisanu funkciju demonstrirajte na brojevima koji se unose sa tastature. Program treba da prekine rad kada se sa tastature unese negativan broj.
2. Napišite funkciju koja ima dva parametra, koji su po tipu nizovi znakova. Za oba parametra se pretpostavlja da predstavljaju nul-terminirane nizove stringove (tj. znak NUL označava kraj stringa). Funkcija treba da kao rezultat vrati 0 ukoliko su nizovi znakova jednaki, 1 ukoliko prvi niz znakova po leksikografskom (abecednom) poretku dolazi iza drugog niza znakova, odnosno -1 ukoliko prvi niz znakova po leksikografskom poretku dolazi ispred drugog niza znakova. Pri tome je zabranjeno korištenje funkcija iz biblioteke **"cstring"**. Drugim riječima, funkcija treba da radi isto kao i funkcija **"strcmp"** iz biblioteke **"cstring"**, samo bez njenog korištenja. Funkciju testirajte na primjerima stringova koji se unose sa tastature.
3. Neki element neke matrice naziva se njeno *sedlo*, ukoliko je on ujedno najveći element u redu u kojem se nalazi i najveći element u koloni u kojoj se nalazi, ili obrnuto (tj. najmanji element u redu u kojem se nalazi i najmanji element u koloni u kojoj se nalazi). Nemaju sve matrice sedlo, a neke imaju i više od jednog. Napišite funkciju koja kao parametar prima matricu formata 5×5 , a kao rezultat vraća logičku vrijednost **"true"** ili **"false"** ovisno od toga da li matrica proslijeđena kao parametar sadrži barem jedno sedlo ili ne. Funkciju testirajte na primjeru matrice čiji se elementi unose sa tastature.