

Zadaci za samostalno vježbanje 2.

Svi zadaci dati ovdje su takvi da se mogu uraditi korištenjem isključivo gradiva prva dva predavanja i ranije stečenog predznanja na predmetu “Osnove računarstva”. Zvezdicom (*) su označeni zadaci koji su se pojavljivali na dosadašnjim ispitima (eventualno uz minorne modifikacije, koje suštinski ništa ne mijenjaju). Tarabom (#) su označeni zadaci koji su u prethodnim generacijama bili zadaci za zadaću. Savjetuju se studenti da koriste razne C++11 novine (poput rangovskih *for*-petlji pri radu sa nizovima i vektorima) koje su rađene na predavanjima gdje god bi im one mogle biti od koristi, ali da također i razmisle kako bi se isti problem mogao riješiti na “stariji” način (tj. bez dodataka uvedenih u C++11).

Napomena: Među zadacima ima dosta sličnih zadataka, tako da je često moguće minornom prepravkom rješenje jednog zadatka prepraviti u rješenje njemu sličnog zadatka.

1. Napišite program koji za prirodan broj n unesen sa tastature ispisuje na ekran formatiranu tablicu množenja za sve brojeve od 1 do n uključivo. Na primjer, ukoliko se unese $n = 5$, ispis na ekran bi trebao izgledati poput sljedećeg:

```
+-----+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |  5  |
+-----+-----+-----+-----+-----+
|  2  |  4  |  6  |  8  | 10  |
+-----+-----+-----+-----+-----+
|  3  |  6  |  9  | 12  | 15  |
+-----+-----+-----+-----+-----+
|  4  |  8  | 12  | 16  | 20  |
+-----+-----+-----+-----+-----+
|  5  | 10  | 15  | 20  | 25  |
+-----+-----+-----+-----+-----+
```

Pretpostavite da je n takav da će čitava tablica stati na ekran i da niti jedan produkt neće imati više od 3 cifre. Za ispis koristite objekat izlaznog toka “`cout`”, a za prilagođavanje širine ispisa manipulator “`setw`”.

2. Napišite funkciju “`UnesiBroj`” bez parametara, koja od korisnika očekuje da unese neki realni broj. Ukoliko je zaista unesen realan broj, funkcija treba da ga vrati kao rezultat iz funkcije. Ukoliko nije unesen broj, funkcija treba ispisati “Neispravan unos. Molimo, pokušajte ponovo.” i ponovo tražiti broj, sve dok se zaista ne unese ispravan broj (prema tome, funkcija ne završava dok se ne unese ispravan broj). Napisanu funkciju iskoristite u programu koji traži da se unesu koeficijenti a , b i c linearne jednačine $ax + b = c$, a koji zatim nalazi i ispisuje njeno rješenje, pod uvjetom da ono postoji, ili odgovarajuću poruku u suprotnom.
3. Napišite funkciju koja ima jedan cjelobrojni parametar. Funkcija treba da vrati kao rezultat logičku vrijednost “`true`” ukoliko je broj simetričan, odnosno ukoliko se isto čita sa obje strane (na primjer, broj 13431 je simetričan). U suprotnom, funkcija vraća kao rezultat logičku vrijednost “`false`”. Napisanu funkciju trebate demonstrirati u kratkom testnom programu na brojevima koji se unose sa tastature. Program za svaki uneseni broj treba da ispise da li je simetričan ili nije. Program treba da završi rad kada se kao broj unese 0.
- 4*. Napišite funkciju koja kao parametar prima neki prirodan broj i koja vraća logičku vrijednost “tačno” ili “netačno” u ovisnosti da li u broju koji je proslijeđen kao parametar ima jednakih cifara ili ne. Napišite i kratki testni program u kojem ćete demonstrirati kako se poziva napisana funkcija i kako se može iskoristiti njen rezultat.
- 5*. Napišite funkciju koja kao parametar prima neki prirodan broj i koja vraća logičku vrijednost “tačno” ili “netačno” u ovisnosti da li su u broju koji je proslijeđen kao parametar sve cifre različite ili ne. Napišite i kratki testni program u kojem ćete demonstrirati kako se poziva napisana funkcija i kako se može iskoristiti njen rezultat.
6. 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.

7. Napišite funkciju “Podbroj” sa dva parametra “*n*” i “*parnost*”, od kojih je prvi cjelobrojnog a drugi logičkog tipa. Ukoliko parametar “*parnost*” ima vrijednost “*true*”, funkcija treba da kao rezultat vrati broj koji se sastoji samo od *parnih* cifara broja “*n*” (uzetih u istom poretku), a ukoliko parametar “*parnost*” ima vrijednost “*false*”, funkcija treba da kao rezultat vrati broj koji se sastoji samo od *neparnih* cifara broja “*n*”. Na primjer, nakon izvršenja naredbi

```
std::cout << Podbroj(3427816, true) << std::endl;  
std::cout << Podbroj(3427816, false) << std::endl;
```

treba da se ispišu brojevi 4286 i 371. Napisanu funkciju trebate demonstrirati na kratkom testnom programu koji će za broj unesen sa tastature ispisati brojeve sastavljene od njegovih parnih odnosno neparnih cifara. Proces treba da se ponavlja sve dok korisnik ne unese nulu.

8. Neka je *n* prirodan broj, i neka je *n*₁ proizvod cifara broja *n*, *n*₂ proizvod cifara broja *n*₁, *n*₃ proizvod cifara broja *n*₂ itd. Najmanji broj *k* za koji vrijedi da je *n*_{*k*} jednocifren broj u teoriji brojeva se naziva *multiplikativna otpornost* broja *n*. Na primjer, multiplikativna otpornost broja 6788 iznosi 6, jer je:

$$\begin{aligned}n_1 &= 6 * 7 * 8 * 8 = 2688 \\n_2 &= 2 * 6 * 8 * 8 = 768 \\n_3 &= 7 * 6 * 8 = 336 \\n_4 &= 3 * 3 * 6 = 54 \\n_5 &= 5 * 4 = 20 \\n_6 &= 2 * 0 = 0\end{aligned}$$

Napišite funkciju “*MultOtpornost*” koja kao parametar prima prirodan broj *n* a vraća kao rezultat njegovu multiplikativnu otpornost.

Napisanu funkciju trebate demonstrirati na kratkom testnom programu koji će ispisati sve brojeve u opsegu od *a* do *b* uključivo koji imaju zadanu multiplikativnu otpornost *k* pri čemu se vrijednosti *a*, *b* i *k* zadaju putem tastature. Na primjer, ukoliko se za *a*, *b* i *k* unesu vrijednosti 9985, 10002 i 3, treba da se ispišu brojevi 9987, 9988, 9989, 9992, 9993, 9994, 9995, 9998 i 9999, jer ti brojevi u zadanom intervalu imaju multiplikativnu otpornost 3.

Napomena: Za sve razumne vrijednosti *n*, multiplikativna otpornost ima uglavnom jednocifrenu vrijednost, pa o tome vodite računa kada budete testirali program. Brojevi koji imaju multiplikativnu otpornost veću od 9 izrazito su rijetki!

9. Napišite program koji traži da se sa tastature unese jedan kompleksan broj *z* i dva realna broja *x* i *y*, a koji zatim računa i ispisuje vrijednost izraza

$$\left[x + \frac{5i}{z^3 \sqrt{x-yi}} \right]^2$$

Na primjer, ukoliko se sa tastature unesu podaci *z* = (3, 7), *x* = 2 i *y* = 6, program treba da ispiše rezultat (4.004757, -0.0173699) jer je

$$\left[2 + \frac{5i}{(3+7i)^3 \sqrt{2-6i}} \right]^2 = 4.004757 - 0.0173699i$$

10. Poznata Moievrova teorema za kompleksne brojeve tvrdi da je za ma kakve realne brojeve *x* i *n*

$$(\cos x + i \sin x)^n = \cos nx + i \sin nx$$

Napišite program koji će za realne brojeve *x* i *n* unesene sa tastature izračunati lijevu i desnu stranu ovog izraza i ispisati ih na ekranu, koristeći kompleksni tip podataka. Na nekoliko primjera provjerite da li se zaista dobijaju jednake vrijednosti.

11. Prepravite program za rješavanje kvadratne jednačine rađen na Predavanju 2. koji koristi kompleksne promjenljive tako da dopusti da i koeficijenti kvadratne jednačine budu kompleksni brojevi. Pri tome, rješenja čiji je imaginarni dio jednak nuli treba ispisati tako da se ispisuju čisto kao realni brojevi, a ne kao uređeni parovi realnih brojeva. Uputa: koristite funkciju “*real*”.

12. Napišite funkcije `"SviElementiSuJednaki"` i `"SviElementiSuRazliciti"`. I jedna i druga funkcija primaju kao parametar jedan vektor realnih brojeva, a vraćaju kao rezultat logičku vrijednost (tj. vrijednost tipa `"bool"`). Funkcija `"SviElementiSuJednaki"` vraća vrijednost `"true"` ako i samo ako su svi elementi vektora međusobno identični, a u suprotnom vraća `"false"`. Funkcija `"SviElementiSuRazliciti"` vraća vrijednost `"true"` ako i samo ako su svi elementi vektora međusobno različiti (tj. ukoliko nikoja dva elementa nisu međusobno jednaka), a u suprotnom vraća `"false"`. Napisane funkcije testirajte u glavnom programu (`"main"` funkciji) u kojem se prvo sa tastature unosi prirodan broj n i n elemenata vektora, a nakon toga pozivaju napisane funkcije sa ciljem da se utvrdi da li su svi elementi vektora jednaki odnosno različiti (naravno, rezultate obavljene analize treba prikazati na ekranu u vidu odgovarajućih poruka).
- 13*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja vraća logičku vrijednost `"tačno"` ili `"netačno"`, u ovisnosti da li u vektoru ima parnih brojeva ili ne. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi prirodan broj n a zatim n cijelih brojeva, koji se unose u vektor. Nakon unosa, program treba pozvati napisanu funkciju sa ciljem da utvrdi ima li među unesenim brojevima parnih brojeva ili ne i ispisati odgovarajuću poruku, ovisno od rezultata funkcije.
- 14*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja vraća logičku vrijednost `"tačno"` ili `"netačno"`, u ovisnosti da li u vektoru ima potpunih kvadrata (tj. brojeva koji se mogu napisati kao kvadrati nekog drugog prirodnog broja) ili ne. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi prirodan broj n a zatim n cijelih brojeva, koji se unose u vektor. Nakon unosa, program treba pozvati napisanu funkciju sa ciljem da utvrdi ima li među unesenim brojevima potpunih kvadrata ili ne i ispisati odgovarajuću poruku, ovisno od rezultata funkcije.
- 15*. Napišite funkciju koja ima dva parametra, od kojih je prvi niz realnih brojeva, a drugi broj elemenata u nizu. Funkcija treba da kreira vektor realnih brojeva koji sadrži isti broj elemenata kao i zadani niz, da prepíše sve elemente niza u kreirani vektor u obrnutom poretku (tj. prvi element niza treba da postane posljednji element vektora, drugi element niza treba da postane pretposljednji element vektora, itd.), i da vrati tako popunjeni vektor kao rezultat. Napisanu funkciju demonstrirajte u glavnom programu koji prvo sa tastature unosi elemente niza kapaciteta 10 elemenata. Nakon toga, program poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi u obrnutom poretku u odnosu na elemente unesenog niza. Na kraju, program ispisuje elemente generiranog vektora na ekran (međusobno razdvojene razmakom).
- 16*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja vraća logičku vrijednost `"tačno"` ili `"netačno"`, u ovisnosti da li u vektoru ima trocifrenih brojeva ili ne. Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi prirodan broj n a zatim n cijelih brojeva, koji se unose u vektor. Nakon unosa, program treba pozvati napisanu funkciju sa ciljem da utvrdi ima li među unesenim brojevima trocifrenih brojeva ili ne i ispisati odgovarajuću poruku, ovisno od rezultata funkcije.
- 17*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja vraća logičku vrijednost `"tačno"` ili `"netačno"`, u ovisnosti da li u vektoru ima brojeva koji su stepeni broja 2 ili ne (ukoliko želite, možete koristiti funkciju `"log"` iz biblioteke `"cmath"`, ali to nije neophodno). Napisanu funkciju demonstrirajte u testnom programu u kojem se sa tastature unosi prirodan broj n a zatim n cijelih brojeva, koji se unose u vektor. Nakon unosa, program treba pozvati napisanu funkciju sa ciljem da utvrdi ima li među unesenim brojevima stepena dvojke ili ne i ispisati odgovarajuću poruku, ovisno od rezultata funkcije.
- 18*. Napišite funkciju koja kao parametar prima jedan prirodan broj, a kao rezultat vraća vektor čiji su elementi svi djelioc broja zadanog kao parametar. Napisanu funkciju demonstrirajte u glavnom programu u kojem se sa tastature unosi prirodan broj n i koji nakon toga poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi djelioc broja n , čije elemente na kraju treba ispisati na ekran (međusobno razdvojene razmakom).
- 19*. Napišite funkciju koja kao parametar prima jedan prirodan broj, a kao rezultat vraća vektor čiji su elementi svi prirodni brojevi koji su potpuni kvadrati (tj. koji se mogu napisati kao kvadrat nekog drugog prirodnog broja), a koji su manji od broja zadanog kao parametar. Napisanu funkciju demonstrirajte u glavnom programu u kojem se sa tastature unosi prirodan broj n i koji nakon toga poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi svi potpuni kvadrati manji od n , čije elemente na kraju treba ispisati na ekran (međusobno razdvojene razmakom).

- 20*. Napišite funkciju koja kao parametar prima jedan prirodan broj, a kao rezultat vraća vektor čiji su elementi svi stepeni broja 2, a koji su manji od broja zadanog kao parametar. Napisanu funkciju demonstrirajte u glavnom programu u kojem se sa tastature unosi prirodan broj n i koji nakon toga poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi svi stepeni dvojke manji od n , čije elemente na kraju treba ispisati na ekran (međusobno razdvojene razmakom).
- 21*. Napišite funkciju koja kao parametar prima jedan prirodan broj, a kao rezultat vraća vektor čiji su elementi svi trocifreni brojevi koji su djeljivi sa brojem koji je zadan kao parametar. Napisanu funkciju demonstrirajte u glavnom programu u kojem se sa tastature unosi prirodan broj n i koji nakon toga poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi svi trocifreni brojevi djeljivi sa n , čije elemente na kraju treba ispisati na ekran (međusobno razdvojene razmakom).
- 22*. Napišite funkciju koja ima dva parametra, od kojih je prvi niz cijelih brojeva, a drugi broj elemenata u nizu. Funkcija treba da kreira vektor cijelih brojeva koji sadrži iste elemente kao i zadani niz, ali tako da prvo idu parni, a zatim neparni brojevi (međusobni poredak parnih odnosno neparnih brojeva treba da bude isti kao i u izvornom nizu), i da vrati tako popunjeni vektor kao rezultat. Na primjer, ukoliko je niz sadržavao redom elemente 3, 5, 2, 7, 6, 4, 1, 8, 5, 9, 4, 3 i 4, vraćeni vektor treba da sadrži redom elemente 2, 6, 4, 8, 4, 4, 3, 5, 7, 1, 5, 9 i 3. Napisanu funkciju demonstrirajte u glavnom programu koji prvo sa tastature unosi elemente niza kapaciteta 10 elemenata. Nakon toga, program poziva napisanu funkciju sa ciljem generiranja vektora čiji su elementi u poretku kako je gore opisano u odnosu na elemente unesenog niza. Na kraju, program ispisuje elemente generiranog vektora na ekran (međusobno razdvojene razmakom).
- 23*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja kao rezultat vraća novi vektor čiji su elementi broj cifara odgovarajućih elemenata vektora koji je zadan kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 32, 459, 72, 8, 24, 771 i 13, funkcija treba da kao rezultat vrati vektor čiji su elementi 2, 3, 2, 1, 2, 3 i 2. Napisanu funkciju demonstrirajte u kratkom testnom programu u kojem se sa tastature prvo unosi prirodan broj n , a zatim n elemenata vektora. Program tada poziva napisanu funkciju sa ciljem da kreira novi vektor koji sadrži brojeve cifara unesenih elemenata i ispisuje elemente novokreiranog vektora na ekran (međusobno razdvojene razmakom).
- 24*. Napišite funkciju koja ima jedan cjelobrojni parametar (nazovimo ga n), koja kao rezultat vraća vektor koji se sastoji od svih prostih faktora broja n , pri čemu se svaki faktor pojavljuje onoliko puta koliko se puta on pojavljuje u faktorizaciji broja na proste faktore. Napišite i mali testni program u kojem ćete demonstrirati kako se napisana funkcija može iskoristiti za ispis prostih faktora broja koji se unosi sa tastature.
- 25*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja kao rezultat vraća novi vektor čiji su elementi sume cifara odgovarajućih elemenata vektora koji je zadan kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 32, 459, 72, 8, 24, 771 i 13, funkcija treba da kao rezultat vrati vektor čiji su elementi 5, 18, 9, 8, 6, 15 i 4. Napisanu funkciju demonstrirajte u kratkom testnom programu u kojem se sa tastature prvo unosi prirodan broj n , a zatim n elemenata vektora. Program tada poziva napisanu funkciju sa ciljem da kreira novi vektor koji sadrži sume cifara unesenih elemenata i ispisuje elemente novokreiranog vektora na ekran (međusobno razdvojene razmakom).
- 26*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja kao rezultat vraća novi vektor čiji su elementi najmanje cifre odgovarajućih elemenata vektora koji je zadan kao parametar. Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 32, 459, 72, 8, 24, 771 i 13, funkcija treba da kao rezultat vrati vektor čiji su elementi 2, 4, 2, 8, 2, 1 i 1. Napisanu funkciju demonstrirajte u kratkom testnom programu u kojem se sa tastature prvo unosi prirodan broj n , a zatim n elemenata vektora. Program tada poziva napisanu funkciju sa ciljem da kreira novi vektor koji sadrži najmanje cifre unesenih elemenata i ispisuje elemente novokreiranog vektora na ekran (međusobno razdvojene razmakom).
- 27*. Napišite funkciju koja kao parametar prima vektor cijelih brojeva, a koja kao rezultat vraća novi vektor u koji su prepisani samo oni elementi primljenog vektora koji su složeni brojevi (tj. koji nisu prosti). Na primjer, ukoliko se funkciji proslijedi vektor čiji su elementi 7, 9, 2, 13, 23, 25, 17 i 91 funkcija treba da vrati kao rezultat vektor čiji su elementi 9, 25 i 91. Napisanu funkciju demonstrirajte u kratkom testnom programu u kojem se sa tastature prvo unosi prirodan broj n , a zatim n elemenata vektora. Program tada poziva napisanu funkciju sa ciljem da kreira novi vektor koji sadrži one elemente unesenog vektora koji su složeni brojevi i ispisuje elemente novokreiranog vektora na ekran (međusobno razdvojene razmakom).

- 28*. Napišite funkciju sa dva parametra, pri čemu je prvi parametar neki vektor cijelih brojeva, a drugi parametar cijeli broj. Funkcija treba da vrati kao rezultat novi vektor čiji su elementi oni elementi vektora zadanog kao prvi parametar čija je suma cifara veća od broja zadanog drugim parametrom. Napišite i mali testni program u kojem ćete demonstrirati kako se upotrebljava napisana funkcija.
- 29*. Za neki element a_i nekog slijeda brojeva kažemo da je *lokalni maksimum* ukoliko je manji od svojih susjeda, tj. ukoliko vrijedi $a_i > a_{i-1}$ i $a_i > a_{i+1}$. Napišite funkciju koja kao parametar prima vektor realnih brojeva, a kao rezultat vraća novi vektor u kojem se nalaze svi lokalni maksimumi vektora koji je prosljeđen kao parametar (odnosno prazan vektor ako lokalnih maksimuma nema). Napišite i kratki testni program u kojem ćete demonstrirati kako se poziva napisana funkcija i kako se može iskoristiti njen rezultat.
- 30*. Za neki slijed brojeva kažemo da je *oscilatoran* ukoliko mu elementi naizmjenično rastu i opadaju ili obrnuto. Na primjer, slijed brojeva 3, 7, 4, 6, 1, 5, 2, 3 je oscilatoran, kao i slijed 3, 1, 5, 4, 7, 2, 3, 2. Napišite funkciju koja prima kao parametar vektor realnih brojeva a koja vraća kao rezultat logičku vrijednost "tačno" ili "netačno" u ovisnosti da li je slijed brojeva pohranjen u vektoru oscilatoran ili ne. Napišite i kratki testni program u kojem ćete demonstrirati kako se poziva napisana funkcija i kako se može iskoristiti njen rezultat.
- 31#. Napišite funkciju "OdstraniDuplikate" i "BrojDuplikata" koja kao parametar primaju vektor cijelih brojeva "v". Funkcija "OdstraniDuplikate" treba da kao rezultat vrati novi vektor u koji će biti prepisani svi elementi vektora "v" izostavljajući duplikate (tj. brojeve koji se ponavljaju). Funkcija "BrojDuplikata" treba da vrati kao rezultat broj brojeva koji se ponavljaju. Na primjer, ukoliko vektor "v" ima elemente 7, 10, 4, 2, 4, 4, 5, 6, 7, 3, 9, 1, 8, 6 i 7, funkcija "OdstraniDuplikate" treba da vrati vektor sa elementima 7, 10, 4, 2, 5, 6, 3, 9, 1 i 8, dok funkcija "BrojDuplikata" treba da vrati broj 3, jer se 3 broja ponavljaju (to su brojevi 7, 4 i 6). Napisane funkcije testirajte u programu koji traži da se prvo unese prirodan broj n , a nakon toga elementi nekog vektora koji ima n cjelobrojnih elemenata. Potom program treba da ispiše elemente vektora iz kojeg su odstranjeni brojevi koji se ponavljaju, kao i broj elemenata koji se ponavljaju.
- 32#. Napišite funkcije "ProstiBrojevi", "ProstiBrojeviBlizanci", "ProstiFaktori" i "Djeliooci". Sve četiri funkcije imaju jedan cjelobrojni parametar (nazovimo ga n). Funkcija "ProstiBrojevi" treba da kreira i kao rezultat vrati vektor koji se sastoji od prvih n prostih brojeva. Funkcija "ProstiBrojeviBlizanci" treba da vrati vektor koji ima $2n$ elemenata, a koji se sastoji od prvih n parova tzv. prostih brojeva blizanaca, koji predstavljaju proste brojeve koji se međusobno razlikuju za 2 (npr. 3 i 5, 11 i 13, 29 i 31, itd.). Funkcija "ProstiFaktori" treba da kao rezultat vrati vektor koji se sastoji od svih prostih faktora broja n , pri čemu se svaki faktor pojavljuje onoliko puta koliko se puta on pojavljuje u faktORIZACIJI broja na proste faktore. Funkcija "Djeliooci" treba da kao rezultat vrati vektor koji se sastoji od svih djelilaca broja n . Napisane funkcije testirajte u malom testnom programu u kojem se sa tastature unosi prirodan broj n , a zatim pozivaju napisane funkcije sa ciljem generiranja odgovarajućih vektora koji se na kraju ispisuju na ekran (uz odgovarajuće poruke koje opisuju šta ti vektori predstavljaju).