

Priprema za jedanaeste labose iz kolegija Programiranje

Argumenti funkcije, reference i rekurzije

Napomena: proučiti materijale vezane uz jedanaesto predavanje

Napišite program u C++ koji će korisniku staviti na raspolaganje izbor sa sljedećim mogućnostima:

1. Sortiranje polja postupkom Bubble sort
2. Površina trokuta
3. Eratostenovo sito
4. Elementi niza
5. Izlaz iz programa

Za 1 bod

Potrebno je korisniku ispisati mogućnosti, te omogućiti unos izbora. Potrebno je definirati statičko polje brojača od točno 5 elemenata za brojanje izvršenih izbora korisnika.

Za izbor mogućnosti treba napisati iteraciju tipa **do-while** unutar koje se izvršava sljedeće:

- a.) poziva se funkcija tipa **int** za izbor mogućnosti (ispisuje korisniku mogućnosti, zatraži unos, te vraća vrijednost 1-5, odnosno 0 ako je izbor neispravan)
- b.) ako je izbor bio ispravan, poziva se funkcija tipa **bool** s argumentom tipa **int** (korisnikov izbor) unutar koje se nalazi selekcija tipa **switch** s pozivima potprograma koji implementiraju pojedine mogućnosti programa (funkcija vraća **true** ako je operacija uspješno izvršena ili **false** ako nije izvršena)
- c.) ako je izbor bio ispravan i operacija uspješno izvršena, povećati odgovarajući brojač za 1, u suprotnom ispisati odgovarajuću poruku (da je izbor bio neispravan ili da operacija nije izvršena).

Na kraju treba ispisati koliko puta je izvršena pojedina mogućnost programa i to pomoću odgovarajućeg potprograma (argument je adresa polja brojača).

Za 2 boda:

Mogućnost 1. Unutar funkcije s pozivima mogućnosti (b.) upitati korisnika za broj elemenata polja (N). Ako je $N < 10$ ili $N > 1000$ ispisati odgovarajuću poruku, a funkcija vraća **false**. Dinamički alocirati polje od N elemenata tipa **float** i generirati njihove vrijednosti u rasponu 50-100 uz pomoć generatora slučajnih brojeva. Pozvati funkciju za sortiranje brojeva silazno postupkom **Bubble** sort (tipa **float**, vraća vrijednost najvećeg elementa polja) s dva argumenta (broj elemenata polja i adresa polja). Funkcija za sortiranje treba najprije ispisati nesortirano polje, a zatim polje sortirati. Sortirano polje i vrijednost najvećeg elementa treba ispisati unutar funkcije s pozivima mogućnosti.

Za 3 boda:

Mogućnost 2. Unutar funkcije s pozivima mogućnosti (b.) upitati korisnika za duljine stranica trokuta **a**, **b** i **c**. Pozvati funkciju za računanje površine trokuta (bez tipa) s četiri varijabilna argumenta (površinom **P**, te duljinama stranica **a**, **b** i **c**). Funkcija kroz argument **P** vraća vrijednost površine trokuta, te se ona ispisuje, zajedno s duljinama stranica. Ako tri stranice ne čine trokut tada sve tri

stranice treba povećati za 1, a funkcija kroz argument **P** vraća vrijednost -1. Funkcija za površinu tada se poziva ponovo, dok ne vrati vrijednost površine, odnosno najviše pet puta nakon čega funkcija s pozivima mogućnosti treba vratiti *false*.

$$s = \frac{a + b + c}{2}$$

$$P = \sqrt{s * (s - a) * (s - b) * (s - c)}$$

Za 4 boda:

Mogućnost 3. Unutar funkcije s pozivima mogućnosti (**b.**) upitati korisnika za donju (**A**) i gornju (**B**) granicu raspona unutar kojeg će se tražiti prosti brojevi postupkom Eratostenovog sita. Obje vrijednosti moraju biti unutar raspona 2..1000, s tim da **B** mora biti veći od **A**. U suprotnom funkcija s pozivima mogućnosti treba vratiti *false*. Dinamički alocirati odgovarajuće polje veličine (**B-A**), pozvati funkciju za Eratostenovo sito (tipa *int* - vraća ukupan broj prostih brojeva u zadanom rasponu; argumenti funkcije su granice intervala **A** i **B**, te adresa polja) te ispisati ukupan broj prostih brojeva u zadanom intervalu.

Eratostenovo sito funkcionira tako da se najprije svi brojevi (=odgovarajući elementi polja) postave na vrijednost 1 (=prosti broj). Nakon toga izbacuju se svi višekratnici brojeva 2 na dalje (=odgovarajući elementi polja postave se na 0) . Na kraju treba ispisati sve proste brojeve u rasponu **A..B**.

Za 5 bodova:

Mogućnost 4. Elementi niza izračunavaju se prema sljedećoj formuli:

$$F_n = F_{(n-1)} + 2 * F_{(n-2)} - F_{(n-3)}$$

$$F_1=1, F_2 = 1, F_3 = 1$$

Primjer:

1, 1, 1, 2, 3, 6, 10, 19, ...

Unutar funkcije s pozivima mogućnosti (**b.**) upitati korisnika za redni broj elementa niza (**n**), zatim pozvati odgovarajuću rekurzivnu funkciju tipa *int* (argument je redni broj elementa) koja vraća vrijednost **n**-tog elementa niza, odnosno vrijednost -1, ako ta vrijednost prelazi najveću moguću vrijednost za varijable tipa *int*. Vrijednost elementa (ili poruku da prelazi najveću vrijednost za varijable tipa *int*) niza ispisati unutar funkcije s pozivima mogućnosti. Postupak se ponavlja tako dugo dok korisnik ne ponovi istu vrijednost **n** tri puta za redom!