Zadatak 1:

*Napišite C++ kôd za sljedeći zadatak:*

Napravite program koji će računati aritmetičku sredinu brojeva od 1 do 100, ali izuzimajući brojeve od 40 do 60.

Znači, potrebno je izračunati aritmetičku sredinu za brojeve:

1, 2, 3, ..., 37, 38, 39, 61, 62, 63, ..., 99, 100

*Pomoć:*

* koristite varijablu s za sumu
* koristite brojač b koji će brojati brojeve koje zadovoljavaju dati uslov
* neka *for*-petlja mijenja svoj brojač i od 1 do 100
* u *for*-petlju dodajte *if*-iskaz koji će, ako je broj trenutni broj i zadovoljio uslov, uvećati brojač i povećati sumu
* kako glasi uslov za *if*-iskaz? kad je broj i zadovoljio uslov?
* vi trebati samo provjeriti da li je trenutno i manje od 40 i veće 60
* provjeru da li je i manje od 1 ili veće od 100 ne trebate vršiti, jer vrijednost i može biti samo između 1 i 100, nikako van toga
* da li trenutno broj i zadovoljio zahtijevani uslov ako vrijedi (i<=40 && i>=60) ?
* nije, zbog dva razloga:

1. Granične vrijednosti ne smijemo uzeti u računanje!

Novi prepravljen uslov glasi:

(i<=39 && i>=61)

1. Ni prethodno napisan uslov nije ispravan. *Zašto?*

Pogledajte taj uslov matematički. Nabrojte nekoliko brojeva koji zadovoljavaju taj uslov!

Npr:

* broj 50 ne zadovoljava uslov jer nije veći od 60, a nije ni manji od 40
* broj 70 ne zadovoljava uslov jer nije manji od 40
* broj 10 ne zadovoljava uslov jer nije veći od 60



Ni jedan broj ne zadovoljava gornji uslov. Znači, trebat ćete uslov „(i<=39 && i>=61)“ malo prepraviti. Pokušajte sami.

Zadatak 2:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko cijelih brojeva. Nakon što je korisnik unio sve te brojeve, program treba ispisati najveći uneseni broj.

*Pomoć:*

* koristit ćemo varijablu m koja će pamtiti koji je do sada najveći uneseni broj
* u *for*-petlju ćemo dodati
  + unos broja x sa tastature
  + if-iskaz koji će, *ako je* m manje od x, varijabli m dodijeliti vrijednost x, *inače* neće se ništa izvršiti

Mi smo do sada brojačima i sumama postavljali početnu vrijednost na nulu. Sada pitanje glasi: *Koja je početna vrijednost varijable m?*

*Prvi način:*

Ako bi se dogovorili da naš program može raditi samo sa brojevima od –10.000 do +10.000, onda bi početna vrijednost varijable m mogla biti –10 001. U tom slučaju bi *for*-petlja radila sljedeće (slijedi primjer):

--------------------------------------------------------------------------------------------------

* korisnik unosi broj x (npr. x=-123) koji bi trebao biti veći od početne vrijednosti
* pošto je m manje od x (-10000 < -123) uslov *if*-iskaza je zadovoljen pa će izvršiti naredba m = x; (m=-123)

--------------------------------------------------------------------------------------------------

* zatim korisnik unosi opet broj x (neka to bude –200)
* uslov m < x (-123 < -200) nije zadovoljen, varijabla m ostaje neizmijenjena

--------------------------------------------------------------------------------------------------

* zatim korisnik unosi opet broj x (neka to bude 33)
* uslov m < x (-123 < 33) je zadovoljen, varijabla m dobija vrijednost 33

--------------------------------------------------------------------------------------------------

…

* na kraju će vrijednost m biti najveći broj od svih unesenih brojeva

*Drugi način:*

Bolje rješenje ovog zadatka bi bilo kada bi početna vrijednost varijable m bila vrijednost prvog unesenog broja x. To možemo učiniti na dva načina:

1. Dodat ćemo još jedan unos sa tastature broja m prije *for*-petlje, a *for*-petlju ćemo ponavljati samo (n-1) puta.

Probajte program riješiti i na ovakav način!

1. Kad se petlja bude **prvi put** izvršavala, varijabli m ćemo dodijeliti vrijednost x. To možemo implemetirati na dva načina:

|  |  |
| --- | --- |
| b1) | Koristit ćemo ***if-else***-iskaz (dvostruki izbor), koji će, ako se petlja prvi put izvršava (i==1):   * varijabli m dodijeliti vrijednost x,   *inače (ako se petlja ne izvršava prvi put):*   * izvršit će se isti *if*-iskaz kao u prethodnim rješenjima |
| b2) | Koristit ćemo ***if-*iskaz** (jednostruki izbor), koji će, ako se petlja prvi put izvršava (i==1):   * varijabli m dodijeliti vrijednost x   Poslije tog *if*-iskaza će se, bez obzira da li se petlja izvršava prvi put ili ne, izvršiti isti *if*-iskaz kao u prethodnim rješenjima. To se smije izvršiti prvi put kada se petlja ponavlja, jer će varijabla m imati neku vrijednost. |

Zadatak 3:

*Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:*

Napravite program koji će zahtijevati unos broja n. Program treba ispisati i izbrojiti sve brojeve od 2 do 10 sa kojima je broj n djeljiv. Program formatirajte na sljedeći način:

Unesi broj n: 18

1. Broj 18 je djeljiv sa 2

2. Broj 18 je djeljiv sa 3

3. Broj 18 je djeljiv sa 6

4. Broj 18 je djeljiv sa 9

Broj 18 je djeljiv sa ukupno puta: 4

*Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.*

*Pomoć:*

* koristite brojač b kojeg trebate povećati svaki put kada broj n bude djeljiv sa i

Zadatak 4:

*Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:*

Napišite program koji će za uneseni broj provjeriti je li riječ o prostom ili složenom broju:

Unesi broj n: 17

Broj je prost.

Unesi broj n: 18

Broj NIJE prost.

*Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.*

*Šta je to prost broj? Pomoć:*

Primjer prostih brojeva: 3, 5, 7, 11, 13, 17, 19, 23, 33, 37, …

Definicija prostog broja glasi:

*Broj* ***n*** *jer prost ako je djeljiv* ***samo*** *sa 1 i sa samim sobom (n).*

U programiranju ne postoji iskaz kojim možemo reći 'samo', ali možemo reći djeljiv ili nije djeljiv sa tim i tim i tim… S toga morat ćemo našu definiciju promijeniti tako da ne koristimo riječ 'samo'. Evo korisnije definicije:

*Broj* ***n*** *je**prost ako NIJE djeljiv ni sa jednim brojem osim sa* ***1*** *i sa* ***n****.*

Ista definicija ali bez riječi 'osim':

*Broj* ***n*** *je prost ako NIJE djeljiv ni sa jednim brojem u intervalu od* ***2*** *do (****n-1****).*

Kako riješiti zadatak?

*Ako je broj* ***n*** *djeljiv makar sa jednim brojem od 2 do (****n-1****), onda on nije prost.*

**Zadatak 5**

Unesite brojeve x i y

2

6

===============

Broj 2 je prost.

Broj 3 je prost.

Broj 4 NIJE prost

Broj 5 je prost.

Broj 6 NIJE prost

**Zadatak 6**

Kode za provjeru jel broj prost prebacite u novu funkciju **bool JelProst(int b)**