Zadaci – 7. dio, *do-while petlja*

Pocetak programa!

1. Brojim unazad: 10

2. Brojim unazad: 9

3. Brojim unazad: 8

4. Brojim unazad: 7

5. Brojim unazad: 6

6. Brojim unazad: 5

7. Brojim unazad: 4

8. Brojim unazad: 3

9. Brojim unazad: 2

10. Brojim unazad: 1

Kraj programa!

Zadatak 71:

Napravite program koji će ispisati poruke kao na slici koristeći *do*-*while*-petlju.

1. brojač (varijabla) i bi se trebao mijenjati od 0 do 9
2. brojač (varijabla) i bi se trebao mijenjati od 10 do 1

*Riješen C++ kôd se nalaze na stranici 107.*

**Zadatak 72:**

*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.

*Rješenje se nalazi 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 da je broj djeljiv ili nije djeljiv sa tim i tim i tim… S toga koristiti ćemo drugi definiciju koja ne sadrži u sebi izraz SAMO:

*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 (barem) sa jednim brojem od 2 do (****n-1****), onda on nije prost.*

Koristit ćemo petlju da bi izbrojili sa koliko brojeva je broj **n** djeljiv.

Ovaj program je sličan prethodnom programu (zadatak 98). U njemu smo brojili sa koliko brojeva (u intervalu od 2 do 10) je broj **n** djeljiv. To ćemo činiti i ovdje ali u intervalu od *2* do (***n-1***). Ako bi takav program ispisao da je broj **n** djeljiv 0 puta, to bi značilo da je broj **prost**, inače, u slučaju da program ispiše 1, 2, 3 ili bilo koji drugi broj osim nule, značilo bi da uneseni broj **n** **nije prost**. I ovakav program bi bio koristan, ali ne bi bio *user-friendly*. S toga trebamo doraditi program, tako da umjesto da ispisivanja svih brojeva iz datog intervala (i odgovarajuće poruke "djeljiv" ili "nije djeljiv"), ispisuje jedna poruka poruka: '…je prost' ili '…NIJE prost'. Iz prethodnog program trebat ćemo još izbaciti nepotrebnu *cout*-naredbu za ispis poruke "Broj … je djeljiv sa …".

Zadatak 73:

Optimizirajte prethodni program:

1. Kao što vidite, zadatak smo riješili tako što smo brojali sa koliko brojeva je djeljiv broj **n**. Međutim, program ne nudi optimalno rješenje. Moguće je smanjiti korištenje resursa za dva puta, tako što ćemo provjeravati djeljivost broja **n** samo sa brojevima iz intervala od **2** do **n/2.** Primjer:

Za n=36597, provjeravat ćemo djeljivost broja **n** samo u intervalu od 2 do 18298, jer broj 36597 nije sigurno djeljiv ni sa jednim brojem iz intervala 18299 do 36596.

1. Dodatno optimiziranje programa:

Ako smo u toku provjeravanja (izvršavanja petlje) našli da je broj **n** djeljiv sa jednim brojem (brojem većim od 1) možemo odmah prekinuti petlju - jer broj **n** sigurno nije prost.

Poslije petlje slijedi *if-else*-iskaz (dvostruki izbor) koji treba, na osnovu nekog uslova, ispisati "…je prost" ili "…nije prost".

Pošto, *if*-*else*-iskaz 'ne zna' da li se petlja prekinula, moramo (morali smo) u *if*-iskazu (n%i==0) promijeniti neku vrijednost neke varijable tako da *if-else*-iskaz koji ispisuje poruku može 'znati' da li se *if*-iskaz (n%i==0) izvršio.

Promjena neke vrijednosti u *if*-iskazu (n%i==0) može da bude: brojac=1 ili prost=false ili sl.

(*Napomena*: varijable brojac ili prost moraju da imaju neku početnu vrijednost prije izvršavanja petlje. Početna vrijednost mora biti različita od vrijednosti *1* za varijablu brojac i različita od vrijednosti *false* za varijablu prost.)

Prekidanje petlje možete vršiti na dva načina:

1. Pomoću dodatnog uslova za prekid petlje.

Ako koristite varijablu brojac, uslov bi glasio:

* + i<=n/2 && brojac!=1 (čita se: ponavlja ako je …)

*dovoljno je da jedan uslov ne bude zadovoljen i petlja se prekida*

* + !(i>n/2 || brojac==1) (čita se: ponavlja ako je …)

*dovoljno je da jedan uslov bude zadovoljen i petlja se prekida*

1. Pomoću naredbe break: Kada se prvi put u petlji izvrši naredba break, petlja se prekida i nastavlja se sa izvršavanjem naredne linije koda koja slijedi nakon petlje.

*Rješenje se nalazi na kraju dokumenta.*

*(*Ako su vam ovi logički uslovi nejasni pročitajte ponovo *Workshop – 6. dio - logički operatori*!)

Zadatak 74:

Napraviti program koji će ispisati sva cjelobrojna rješenja jednačine:

*3x–10y=5* za *y=1 do 60*

*Rješenje se nalazi na kraju dokumenta.*

*Pomoć:*

* trebate varijablu y mijenjati pomoću petlje od 1 do 60
* izračunajte vrijednost x:
* *3x-10y=5 ⇒ *
* ako je x cijeli broj, ispišite x i y (ako vrijednost u brojniku djeljiva sa nazivnikom)

**Zadatak 75:**  *(zadatak sa ispita: UPR - 14.02.2005.)*

Napišite program koji izračunava zbir parnih brojeva od 0 od 1000 koji su djeljivi sa 7 i 3 ili koji su djeljivi sa 8. Program treba, osim rezultata, da ispisuje i brojeve koji zadovoljavaju uslov. Ispis širine 5 mjesta.

*Rješenje se nalazi na kraju dokumenta.*

Pomoć:

* 1. **i%3 == 0** *(djeljiv sa 3)*
  2. **i%7 == 0** *(djeljiv sa 7)*
  3. **i%8 == 0** *(djeljiv sa 8)*
  4. **i%2 == 0** *(parni brojevi)*

**Zadatak 76:**

Napraviti program za skraćivanje razlomka. Program riješiti u skladu sa fazama programiranja!

*Rješenje se nalazi na kraju dokumenta.*

*Pomoć:*

* input: bojnik i nazivnik
* provjerit ćemo koji ima manju vrijednost bojnik ili nazivnik: manju vrijednost ćemo pridružiti varijabli manji (zbog optimizacije programa)
* moramo provjeriti djeljivost varijable brojnik i nazivnik sa brojevima iz opsega [manji, 2] (ako ne želimo optimizirati programa, onda je svejedno da li ćemo koristiti opseg [brojnik, 2] ili opseg [nazivnik, 2] )
* ako su brojnik i nazivnik djeljivi sa trenutnom vrijednošću brojača petlje (i), podijeliti ćemo ih sa brojem i
* output (skraćeni razlomak): varijabla brojnik i varijabla nazivnik

# Rješenja

|  |  |
| --- | --- |
| Rješenje zadatka br. **71 - a**: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19: | #include <iostream>  using namespace std;  void main()  {  cout << "Pocetak programa! \n";  int i=0;  do  {  cout << i+1 <<". Brojim unazad: " << 10-i << endl;  i++;  }while (i<=9);  cout << "Kraj programa! \n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **71 - b**: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16: | void main()  {  cout << "Pocetak programa! \n";  int i=10;  do  {  cout << 11-i <<". Brojim unazad: " << i << endl;  i--;  }while (i>=1)**;**  cout << "Kraj programa! \n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **72** *(pomoću do-while-petlja)*: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27: | void main()  {  int n, brojac;  cout << "Unesi broj: ";  cin >> n;  brojac = 0;    int i=2;    do  {  if (n%i == 0)  brojac++;  i++;  }while (i<=n/2);  if (brojac == 0)  cout << "Broj je prost\n";  else  cout << "Broj nije prost\n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **73 (1)** *(optimiziran program uz pomoć dodatnog uslova za prekid)*: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26: | void main()  {  int n;  bool prost;  cout << "Unesi broj: ";  cin >> n;  prost = true; // broj je prost (nevin) dok mu se ne dokaže da je složen (kriv)  int i=2;  do  {  if (n%i == 0)  prost=false; // dokazano da je broj slozen – dokazana je krivica  i++;  }while (i<=n/2 && prost==true);  if (prost) // if (prost ==false)  cout << "Broj je prost\n";  else  cout << "Broj nije prost\n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **73 (2)** *(optimiziran program uz pomoć naredbe break)*: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27:  28:  29: | void main()  {  int n;  bool prost;  cout << "Unesi broj: ";  cin >> n;  prost = true;  int i=2;    do  {  if (n%i == 0)  {  prost=false; // ova se naredba mora prije prekida izvrsiti  break; // prekid - izlaz iz petlje  }  i++;  }while (i<=n/2);  if (prost) // if (prost ==false)  cout << "Broj je prost\n";  else  cout << "Broj nije prost\n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **74**: | |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15: | void main()  {  int brojnik;  int y=1;  do  {  brojnik = 5 + 10\*y;  if (brojnik%3 == 0)  {  int x = brojnik/3;  cout << "x = " << x;  cout << " za y = " << y << endl;  }  y++;  }while(y<=60); |

|  |  |
| --- | --- |
| Rješenje zadatka br. **75**: | |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23: | #include <iostream>  #include <iomanip> // biblioteka za funkciju setw, koja omogućava ispis širine 5 mjesta  using namespace std;  void main()  {  int zbir;  zbir = 0;  int i=0;  do  {  if ( (i%2==0) && ((i%3==0) && (i%7==0)) || (i%8 == 0) )  {  cout << i << setw(5);  // koja je razlika između setw(5) i pet space-razmaka?    zbir = zbir + i;  }  i++;  }while(i<=1000);  cout << "\nZbir: " << zbir << endl;  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **76**: | |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27:  28:  29:  30:  31:  32:  33: | void main()  {  int brojnik, nazivnik, manji;  cout << "Unesi brojnik i nazivnik \n";  cin >> brojnik >> nazivnik;    if (nazivnik == 0)  cout << "Greska...\n";  else  {  if (brojnik < nazivnik)  manji = brojnik;  else  manji = nazivnik;  int i=manji;  do  {  if ((brojnik%i==0) && (nazivnik%i==0))  {  brojnik = brojnik / i;  nazivnik = nazivnik / i;  }  i--;  }while(i>=2);  cout << brojnik << " / " << nazivnik << endl;  }  } |