Zadaci – 5a (*if-* i *if-else* iskazi)

## Algoritam predstavljen blok dijagramom

U narednoj tabeli dati su osnovni elementi tok dijagrama:

|  |  |  |  |
| --- | --- | --- | --- |
| Naziv | Algoritam (blok dijagram) | ***C++ code*** | ***Opis*** |
| ulaz |  | **cin** >> a >> b; | vrijednost se unosi sa tastature |
| izlaz |  | **cout** << "Greska"; | ispisuje se poruka na ekran |
|  | **cout** << a << endl << b; | ispisuje se vrijednost varijabli a i b na ekran |
| izraz |  | V = a \* b; | vrijednost (matematičkog izraza) se dodjeljuje varijabli v |
| *if*-iskaz (jednostruki izbor)  *- opšti primjer* - |  | **if** (uslov)  {   |  |  | | --- | --- | | naredba1;  naredba2;  ... | **Blok** |   } | ako je uslov zadovoljen izvršava se Blok (naredbe), inače se preskače Blok |
| *if*-iskaz (jednostruki izbor)  *- konkretan primjer -* |  | **if** (a>0)  {   |  |  | | --- | --- | | b = a\*a;  cout << b;  ... | **Blok** |   } |  |
| *if-else*-iskaz (dvostruki izbor)  *- opšti pimjer* - |  | **if** (uslov)  {   |  |  | | --- | --- | | naredba\_1a;  naredba\_2a;  ... | **Blok A** |   }  **else**  {   |  |  | | --- | --- | | naredba\_1b;  naredba\_2b;  ... | **Blok B** |   } | ako je uslov zadovoljen izvršava se *Blok A*, inače se izvršava *Blok B* |
| *if-else*-iskaz (dvostruki izbor)  *- konkretan primjer* - |  | **if** (a>0)  {   |  |  | | --- | --- | | b = 2\*a;  cout << b;  ... | **Blok A** |   }  **else**  {   |  |  | | --- | --- | | b = -2\*a;  cout << b;  ... | **Blok B** |   } |  |

**Zadatak 31:**

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak (tj.* ***nemojte pisati C++ kôd****):*

Napravite program u kome ćete od korisnika zahtijevati unos dva broja. Neka to budu varijable a i b tipa *integer*. Program treba, ako je prvi broj veći od drugog, ispisati poruku: 'prvi broj je veći od drugog', inače treba ispisati 'prvi broj nije veći od drugog'. Koristite jedan *if-else* iskaz (dvostruki *if*-iskaz). Na kraju, program treba ispisati poruku 'Kraj programa' pomoću naredbe koja se nalazi izvan *if-else* iskaza.

*Rješenje se nalazi na stranici 43.*

*Napomena:*

U dijagramu toka algoritma koji predstavlja rezultat faze dizajna nije potrebno navoditi deklaracije varijabli. Varijable se deklarišu u programu a na osnovu kvalifikacije podataka u fazi dizajna. Za razliku od *source* koda u kome, prije prve upotrebe neke varijable, moramo deklarisati varijablu i tek tada joj možemo dodijeliti neku vrijednost (npr. int a = 5;), u algoritmu ćete odmah koristiti varijablu. Npr. nacrtate jedan pravougaonik i u njega napišete taj izraz (a = 5). Možete zaključiti, da u algoritmu ne navodimo o kojem se tipu podatka radi.

Ako na osnovu algoritma (a nemate ostale elemente iz faze dizajna) želite napisati *source code* onda ćete na osnovu logike morati odabrati koji je tip podatka za neku varijablu (koja je korištena u algoritmu, npr. godiste = 84) najpogodniji (npr.: int godiste = 84;).

U dijagramu toka algoritma se, takođe, **ne crtaju** 'pozdravne' *cout*-poruke, kao: „Zdravo, Unesite broj.“ ili sl., jer se podrazumijevaju i nepotrebne su u dijagramu toka.

U prvih nekoliko zadataka sa *if*-iskazima rješavajte samo pomoću dijagrama toka. Kasnije će, nakon što dobro savladate dijagram toka, slijediti zadaci u kojima ćete pisati C++ kôd na osnovu prethodno nacrtani dijagrama. Primjetit ćete da najteži dio u rješavanju težih zadataka predstavlja crtanje ispravnog dijagram toka.

Zadatak 32:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak (tj. nemojte pisati C++ kôd):*

Prepravite prethodni program tako da se naredba za ispis poruke 'Kraj programa' nalazi u *if-else*-iskazu, tj. jedna naredba za ispis u slučaju 'DA' i jedna naredba za ispis u slučaju 'NE'.

*Rješenje se nalazi na stranici 43.*

Zadatak 33:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak (tj. nemojte pisati C++ kôd):*

Prepravite prethodni program tako da umjesto jednog *if-else*-iskaza (dvostrukog) koristite dva *if*-iskaza (jednostruka), Naredba za ispis neka se nalazi unutar *if*-iskaza.

*Rješenje se nalazi na stranici 43.*

Zadatak 34:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak:*

Napravite program koji će računati vrijednost *y = f(x)* i ispisati je na ekran. Vrijednost x se trebati učitati sa tastature od korisnika.

Primjer rada programa:

*unos* x = -8.  *ispis* y = 3

*unos* x = -0.  *ispis* y = 0

*unos* x = -4.  *ispis* y = 16



Program riješite pomoću tri *if*-iskaza (jednostruki *if*-iskaz). Sva tri iskaza trebaju biti povezani jedan ispod drugog.

Ako ispravno postavite uslove onda će se tačno jedan *if*-iskaz izvršiti jer, za bilo koju vrijednost koju varijable x može imati, uvijek će biti zadovoljen jedan od tri uslova.

*Rješenje se nalazi na stranici 43.*

Zadatak 35:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak:*

Prepravite prethodni program tako da umjesto tri *if*-iskaza koristite dva *if-else*-iskaza.

*Rješenje se nalazi na stranici 44.*

*Mala pomoć:*

U prvi *if-else*-iskaz upišete neki od tri uslova. Neka to bude x>0. Ako je uslov zadovoljen izvršit će se izraz .

A šta će se izvršiti ako uslov nije zadovoljen? – Trebamo izračunati y. Mi još ne znamo po kojoj ćemo formuli izračunati vrijednost y. Tu situaciju možemo prikazati:

|  |
| --- |
| *Napomena*:  isprekidane crtice na ovoj slici ne predstavljaju dijagram toka već nedostatak u dijagramu, tj. nedovršeni dio |

*Kako riješiti Blok X ?*

Blok X treba…

1. …ako je x<0 izvršiti naredbu: y = x2
2. …ako je x=0 izvršiti naredbu: y = 0

Koju će naredbu izvršiti Blok X ako je x>0 ?

* ovo je nemoguć slučaj, jer će se Blok X izvršavati samo ako je x<=0, jer se o tome brine prvi *if*-iskaz

Znači, u Blok X treba ubaciti jedan *if-else*-iskaz sa nekim od gore dva navedena uslova (uslov a i uslov b). Neka to bude x=0. Ostatak je jasan:

|  |  |
| --- | --- |
|  | Potrebno je dopuniti blok dijagram algoritma ovim dijelom koji predstavlja Blok X i tako ga kompletirati. |

Zadatak 36:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak:*

Prepravite prethodni program tako da računa vrijednost y po slijedećoj matematičkoj formuli:

Primjer 1

*unos* x = -8.  *ispis* y = 3

Primjer 2

*unos* x = -0.  *unos* z =-8.  *ispis* y = 11.313

Primjer 3

*unos* x = -4.  *ispis* y = 16



Ako je korisnik za x unio broj 0, program treba samo tada tražiti unos i broja za z.

Dodatni zahtjev: Ako korisnik unese broj koji nije pozitivan (tj. da je manji od nule ili jednak nuli) program treba ispisati poruku: 'Upozorenje: Niste unijeli pozitivan broj.'

*Rješenje se nalazi na stranici 44.*

Zadatak 37:

*Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak:*

Napravite program koji će tražiti od korisnika da unese broj n. Ako je n paran broj (tj. djeljiv sa 2), program će zahtijevati unos boja m. Program treba, u tom slučaju izračunati vrijednost  i ispisati je na ekran. A ako je broj n neparan, program treba izračunati vrijednost…



…i ispisati tu vrijednost na ekran.

*Napomena:*

Program treba, od korisnika, tražiti unos brojeva a i b samo ako bude potrebe.

Pazite! U zadatku nije rečeno, ali se podrazumijeva, da vrijednost m mora biti različita od nule jer nije moguće dijeliti sa nulom. U slučaju da je broj m jednak nuli program treba ispisati neku poruku, kao: 'Greška: Nije moguće dijeliti sa 0' ili sl.

*Rješenje se nalazi na stranici 44.*

*Mala pomoć:*

Kad je broj p djeljiv sa brojem q ?

- Ako je vrijednost ostatka kod dijeljenja brojeva p i q jednaka 0.

Kako izračunati vrijednost ostatka kod dijeljenja dva broja?

- Pogledajte zadatak 30 i zaključite:

* Ako je vrijednost n%2 jednaka 0, onda je n paran broj.
  + inače (ako je vrijednost n%2 različita od 0) je n neparan broj.
* Ako je vrijednost n%2 jednaka 1, onda je n neparan broj.
  + inače (ako je vrijednost n%2 različita od 1) je n paran broj.

Zadatak 38:

Pomoću tabele (na strani 30) pretvorite riješeni algoritam za zadatak br. 31 u C++ kod.

*Rješenje se nalazi na stranici 45.*

Zadatak 39:

Pretvorite riješeni algoritam za zadatak br. 32 u C++ kod.

*Rješenje se nalazi na stranici 45.*

Zadatak 40:

Pretvorite riješeni algoritam za zadatak br. 33 u C++ kod.

*Rješenje se nalazi na stranici 46.*

Zadatak 41:

Pretvorite riješeni algoritam za zadatak br. 34 u C++ kod.

*Rješenje se nalazi na stranici 46.*

Zadatak 42:

Pretvorite riješeni algoritam za zadatak br. 35 u C++ kod.

*Rješenje se nalazi na stranici 47.*

Za pisanje koda sa ugniježđenim *if*-iskazima ( if-iskaz u if-iskazu) preporučujemo da prvo napišite prazan osnovni *if*-iskaz (onaj čiji će uslov biti prvi provjeren) sa praznim vitičastim zagradama za oba slučaja („DA“ i „NE“), a zatim da pomjerite kursor u *if*-iskaz i u njega pišete naredbe (npr.: cout- i cin-naredbe, drugi *if*-iskazi, izrazi, …).

Preporučujemo da sve naredbe u C++ kodu budu uvučene (pomoću tabulatora) u odnosu na vitičaste zagrade kojoj pripadaju:

|  |
| --- |
| if (uslov)  {  naredba\_1;  naredba\_2;  if (uslov)  {  naredba\_3;  }  }  else  {  naredba\_4;  } |

„*if*“, njegov „*else*“ i njihove vitičaste zagrade (ako ih imaju) trebale bi biti podjednako poravnate sa lijeve strane.

Zadatak 43:

Pretvorite riješeni algoritam za zadatak br. 36 u C++ kod.

*Rješenje se nalazi na stranici 47.*

Zadatak 44:

Pretvorite riješeni algoritam za zadatak br. 37 u C++ kod.

*Rješenje se nalazi na stranici 48.*

Rješenja

|  |  |  |
| --- | --- | --- |
| Rješenje zadatka br. **31**: | | Rješenje zadatka br. **32**: |
| Rješenje zadatka br. **33**: | Rješenje zadatka br. **34**: | |

|  |  |
| --- | --- |
| Rješenje zadatka br. **35**: | Rješenje zadatka br. **36**: |
| Rješenje zadatka br. **37**: | |

Rješenje zadatka br. **38** (31):

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20: | #include <iostream>  using namespace std;  void main()  {  int a, b; // deklaracija varijabli se ne crta u algoritmu  cout << "Unesite dva broja \n"; // pozdravna poruku se ne crta u algor.  cin >> a >> b;  **if** (a > b) // ovdje se **ne stavlja** tačka-zarez!  {  cout << "Prvi broj je veci od drugog \n";  }  **else** // ovdje se **ne stavlja** tačka-zarez!  {  cout << "Prvi broj nije veci od drugog \n";  }  cout << "Kraj programa \n";  } |

Ako se u bloku nalazi samo jedna naredba, onda se mogu izostaviti vitičaste {} zagrade. To smo mogli učinit i u prethodnom rješenju:

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16: | #include <iostream>  using namespace std;  void main()  {  int a, b;  cout << "Unesite dva broja \n";  cin >> a >> b;  **if** (a > b)  cout << "Prvi broj je veci od drugog \n";  **else**  cout << "Prvi broj nije veci od drugog \n";    cout << "Kraj programa \n";  } |

Rješenje zadatka br. **39** (32):

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20: | #include <iostream>  using namespace std;  void main()  {  int a, b;  cout << "Unesite dva broja \n";  cin >> a >> b;  if (a > b)  {  cout << "Prvi broj je veci od drugog \n";  cout << "Kraj programa \n";  }  else  {  cout << "Prvi broj nije veci od drugog \n";  cout << "Kraj programa \n";  }  } |

Rješenje zadatka br. **40** (33):

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21: | #include <iostream>  using namespace std;  void main()  {  int a, b;  cout << "Unesite dva broja \n";  cin >> a >> b;  if (a > b)  {  cout << "Prvi broj je veci od drugog \n";  cout << "Kraj programa \n";  }    if (a <= b)  {  cout << "Prvi broj nije veci od drugog \n";  cout << "Kraj programa \n";  }  } |

Rješenje zadatka br. **41** (34):

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  float x, y;  cout << "Unesite x: ";  cin >> x;  if (x > 0)  y = sqrt(x+1);  if (x == 0)  y = 0;  if (x < 0)  y = x \* x;  cout << "y = " << y << endl;  } |

Rješenje zadatka br. **42** (35):

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  float x, y;  cout << "Unesite x: ";  cin >> x;  if (x > 0)  y = sqrt(x+1);  else  {  if (x == 0)  y = 0;  else  y = x \* x;  }  cout << "y = " << y << endl;  } |

Rješenje zadatka br. **43** (36):

|  |  |
| --- | --- |
| 1:  2:  3:  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: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  float x, y;  cout << "Unesite x: ";  cin >> x;  if (x > 0)  y = sqrt(x+1);  else  {  cout << "Upozorenje: Broj nije pozitivan" << endl;  if (x == 0)  {  int z;  cout << "Unesite z: ";  cin >> z;  y = z \* sqrt(float(2));  }  else  y = x \* x;  }  cout << "y = " << y << endl;  } |

Rješenje zadatka br. **44** (37):

|  |  |
| --- | --- |
| 1:  2:  3:  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:  34:  35:  36:  37:  38:  39:  40:  41:  42:  43:  44:  45:  46: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  int n;  cout << "Unesite n: ";  cin >> n;    if (n % 2 == 0)  {  int m; // u zadatku nije rečeno koji tip podatka m mora biti,  cout << "Unesite m: "; // a mi ćemo se odlučiti za int  cin >> m;  if (m != 0)  {  float w; // ovo možemo deklarisati i na početku programa  w = n / float(m);  cout << "w = " << w << endl;  }  else  cout << "Greska: Nije moguce dijeliti sa 0 \n";  }  else  {  float y; // ovo mozemo deklarisati i na početku programa  if (n >= 0)  {  y = sqrt(float(n));  }  else  {  int a, b;  cout << "Unesite a i b: ";  cin >> a >> b;  y = a \* a \* b;  }  cout << "y = " << y << endl;  }  cout << "Kraj programa \n";  } |