Zadaci – 4. dio

Zadatak 27:

Napravite program koji će zahtijevati unos dva broja (a i b) koja će predstavljati dužine kateta pravouglog trougla. Program treba ispisati dužinu stranice c (tj. hipotenuze).

*Rješenje se nalazi na stranici 34.*

*Pomoć:*

Kako izračunati stranicu c?

|  |  |
| --- | --- |
|  |  |

Za računanje korijena koristite funkciju sqrt. Evo primjera:

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11: | #include <iostream>  using namespace std;  #include <math.h> //ova linija se mora dodati zbog funkcije sqrt  void main()  {  float x;  x = sqrt(9.0); // x = korijen iz broja 9,0  cout << x << endl; // ispisat ce se 3  } |

*Pitanje:* Koji tip podatka treba da budu varijable a i b?

*Odgovor*: *int* ili *float*. Ali, ovdje je prikladnije podatak tipa *float*!

*Pitanje:* Koji tip podatka treba da bude varijabla c?

*Odgovor*: Obavezno *float*, jer će vrijednost korijena biti najvjerovatnije decimalni broj!

*Napomena:*

Visual Studio 2003/2005 zahtijeva da argument (parametar) funkcije sqrt bude tipa *float* ili *double* ili *long double*. Stoga nije moguće izračunati korijen iz broja 9, jer je on cijeli broj (*int*), ali moguće je izračunati korijen iz broja 9.0 jer je on decimalni broj.

|  |  |
| --- | --- |
| 8: | a = sqrt(9); // Visual Studio 2005 će prijaviti grešku |

Nije moguće ni izračunati korijen iz varijable koja je tipa *int*:

|  |  |
| --- | --- |
| 0:  0:  0: | int x = 9;  float a;  a = sqrt (x); // Visual Studio 2005 će prijaviti grešku |

A nije ni moguće izračunati korijen iz izraza koji je cjelobrojan:

|  |  |
| --- | --- |
| 0:  0:  0:  0: | int x = 9;  int y = 10;  float a;  a = sqrt (x+y); // Visual Studio 2005 će prijaviti grešku |

*Visual Studio* 6 i stari *Borland C++* ne zahtijevaju da parametar funkcije sqrt bude realan broj (*float* ili sl.), nego on može biti i cijeli broj (*int* ili sl.)

Zadatak 28:

Imate li ideju kako možete ispisati vrijednost korijena iz varijable a, tako da varijabla a ostane tipa *int* ? Riječ je *Visual Studio*-u *2005*!

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  float r;  int a;  cout << "Unesite cijeli broj: ";  cin >> a;    r = sqrt(a); // error C2668: 'sqrt' : ambiguous call to overloaded function  cout << r << endl;  } |

*Rješenje se nalazi na stranici 34.*

Zadatak 29:

Riješite ponovo prethodni zadatak tako da ne koristite dodatnu varijablu!

*Rješenje se nalazi na stranici 34.*

*Mala napomena:*

Iz matematike znate da ne postoji korijen iz negativnih brojeva (barem ne u okviru realnih brojeva), tako je i funkcija sqrt definisana samo za 0 i pozitivne brojeve. Pa ako pokušate izračunati korijen iz negativnog broja program će vam prijaviti grešku dok bude bio pokrenut (tj. u *run-time*-u), bez obzira što je on uspješno kompajliran. Kakva i na koji način će vam se greška prikazati zavisi od toga u kojem razvojnom alatu (Borland C++, Visual C++, ...) je vaš program kompajliran.

Zadatak 30:

Napravite program koji će zahtijevati unos dva cijela broja. Neka do budu varijable a i b tipa ***integer***. Pa ako je korisnik npr. unio brojeve 7 (za a) i 3 (za b) program treba da ispišite rezultate dijeljenja u sljedećem obliku:

|  |
| --- |
| Unesi broj a: 7  Unesi broj b: 3  decimalno dijeljenje: 7:3 = 2.33333333  cjelobrojno odjeljenje: 7:3 = 2  odjeljenje sa ostatkom: 7:3 = 2 + ostatak 1 |

Za one koji ne znaju šta je ostatak kod dijeljenja:

|  |  |
| --- | --- |
|  | **ovo je ostatak djeljenja** |

*Mala pomoć:*

Pročitajte samo ukoliko ne budete mogli samostalno riješiti zadatak:

Da analiziramo koje ćemo varijable koristiti u ispisu:

|  |
| --- |
| Unesi broj a: ...  Unesi broj b: ...  decimalno dijeljenje: a:b = x1  cjelobrojno dijeljenje: a:b = x2  dijeljenje sa ostatkom: a:b = x2 + ostatak x3 |

Sljedeća tabela prikazuje tipove podataka. Ona predstavlja **dizajn**, tj. drugu fazu u kreiranju programa.

|  |  |  |  |
| --- | --- | --- | --- |
| **Podatkovni objekt** | **Vrsta vrijednosti** | **Tip objekta** | **Ime objekta** |
| broj a | varijabla | int | a |
| broj b | varijabla | int | b |
| količnik decimalnog dijeljenja | varijabla | float | x1 |
| količnik cjelobrojnog dijeljenja | varijabla | int | x2 |
| ostatak dijeljenja | varijabla | int | x3 |

*Kako ćemo izračunati* x2*?*

Jednostavno. Vrijednost a/b predstavlja cijeli broj jer se su varijable a i b tipa *int*.

*Kako ćemo izračunati* x3*?*

Jednostavno, koristit ćemo operator za modularno dijeljenje, tj. %. Vrijednost a%b predstavlja ostatak kod dijeljenja brojeva a i b.

*Kako ćemo izračunati* x1*?*

I ovo je jednostavno. Dovoljno je da kod dijeljenja jedna vrijednost bude realna (decimalna) pa će rezultat biti realan. Imamo dva načina:

1. deklarisati ćemo varijablu a1 tipa *float* koja će preuzeti vrijednost varijable a. Zatim ćemo samo još podijeliti a1 i b i dobit ćemo decimalnu vrijednost dijeljenja koju ćemo pridružiti varijabli x1
2. pomoću *casting operatora*: podijeliti ćemo vrijednost float(a) i vrijednost b

*Rješenje se nalazi na stranici 35.*

## ( pre/post) inkrement (uvećanje za 1):

Pogledajte sljedeće primjere:

Primjer za (post)inkrement:

|  |  |
| --- | --- |
| **kod:** | **izlaz iz programa:** |
| int a = 20;  cout << "a = " << a << endl;  a++;  cout << "a = " << a << endl; | a = 20  a = 21 |

Primjer za (pre)inkrement:

|  |  |
| --- | --- |
| **kod:** | **izlaz iz programa:** |
| int a = 20;  cout << "a = " << a << endl;  ++a;  cout << "a = " << a << endl; | a = 20  a = 21 |

Primjer za (post)inkrement:

|  |  |
| --- | --- |
| **kod:** | **izlaz iz programa:** |
| int a = 20;  cout << "a = " << a << endl;  cout << "a = " << a++ << endl;  cout << "a = " << a << endl; | a = 20  a = 20  a = 21 |

Primjer za (post)inkrement:

|  |  |
| --- | --- |
| **kod:** | **izlaz iz programa:** |
| int a = 20;  cout << "a = " << a << endl;  cout << "a = " << ++a << endl;  cout << "a = " << a << endl; | a = 20  a = 21  a = 21 |

*Zaključak:*

U sljedećoj tabeli lijeva i desna strana rade istu stvar:

|  |  |
| --- | --- |
| cout << "a = " << ++a << endl; | a = a+1;  cout << "a = " << a << endl; |
| cout << "a = " << a++ << endl; | cout << "a = " << a << endl;  a = a+1; |

Zadatak:

Šta će program ispisati?

|  |  |
| --- | --- |
| 4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16: | void main()  {  int a = 20;  int b;  b = a++;  a++;  cout << a << endl;  cout << b << endl;  cout << ++a << endl;  cout << b++ << endl;  cout << --a << endl;  cout << b-- << endl;  } |

Rješenja

Rješenje zadatka br. **27**:

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16: | #include <iostream>  using namespace std;  #include <math.h>  void main()  {  float a, b, c;  cout << "Unesite stranicu a: ";  cin >> a;  cout << "Unesite stranicu b: ";  cin >> b;  c = sqrt(a\*a + b\*b);  cout << "c = " << c << endl;  } |

Rješenje zadatka br. **28**:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15: | void main()  {  float r, b;  int a;  cout << "Unesite cijeli broj: ";  cin >> a;  b = a;  r = sqrt(b);  cout << r << endl;  } |

Rješenje zadatka br. **29**:

Ako nemate nikakvih ideja kako riješiti zadatak pogledajte sljedeći primjer, pa će vam najvjerovatnije 'sinuti' neka ideja. Ovaj primjer je potpuno isti prethodnom, samo što ovdje koristimo *casting operator*:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15: | void main()  {  float r, b;  int a;  cout << "Unesite cijeli broj: ";  cin >> a;  **b = float(a);**  r = sqrt(b);  cout << r << endl;  } |

......

Konačno rješenje ovog zadatka:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10:  11:  12:  13:  14: | void main()  {  float r;  int a;  cout << "Unesite cijeli broj: ";  cin >> a;    r = sqrt(float(a));  cout << r << endl;  } |

ili:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10:  11:  12:  13:  14: | void main()  {  float r;  int a;  cout << "Unesite cijeli broj: ";  cin >> a;    **r = sqrt((float)a);**  cout << r << endl;  } |

Rješenje zadatka br. **30**:

*Drugi način:*

|  |  |
| --- | --- |
| 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;  void main()  {  int a, b, x2, x3;  float x1;  cout << "Unesi broj a: ";  cin >> a;  cout << "Unesi broj b: ";  cin >> b;  x2 = a / b;  x3 = a % b;  x1 = float(a) / b;  cout << endl;  cout << "decimalno djeljenje: \t" << a << ":" << b << " = " << x1 << endl;  cout << "cjelobrojno djeljenje: \t" << a << ":" << b << " = " << x2 << endl;  cout << "djeljenje sa ostatkom: \t" << a << ":" << b << " = " << x2 <<  " + ostatak " << x3 << endl;  } |

*Prvi način:*

Umjesto linije br. 16 možemo staviti:

|  |  |
| --- | --- |
| 16:  17: | float a1 = a;  x1 = a1 / b; |