**Zadaci – 10. dio, *Funkcije (drugi dio)***

***(Pozvana) funkcija poziva drugu funkciju***

**Zadatak 99:**

Pogledajte ponovo zadatak broj 110. Da se podsjetimo, zadatak je glasio:

Napravite program koji će od korisnika zahtijevati unos dva broja, m za indeks početka niza i n za indeks kraja niza. Funkcija main treba pozvati funkciju f1 za svaki cijeli broj iz tog raspona [m, n]. Funkcija f1 treba provjeriti da li je vrijednost koju ona prima kvadrat nekog broja. Samo ako jeste, funkcija ga treba ispisati na ekran.

Implementacija definicije funkcije f1 nam nije sada bitna. Obratite pažnju na poziv i prototip funkcije f1.

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22: | #include <iostream>  #include <math.h>  using namespace std;  void f1(int);  void main()  {  int m, n;  cout << "Unesi pocetak i kraj niza: \n";  cin >> m >> n;  for (int i=m; i<=n; i++)  {  f1(i); // ovu funkciju pozivamo (n-m+1) puta  }  }  void f1(int u1)  {  ... |

Prepravite gornji program slijedeći navedene zahtjeve:

* definiciju funkcije f1 nemojte mijenjati
* *for*-petlja sa pozivom funkcije f1 se mora sada nalaziti u novoj funkciji f0
* funkciju f0 pozovite samo jednom iz funkcije main

**Unesi pocetak i kraj niza:**

**2**

**33**

**----Ulazak u funkciju f0----**

**4 = 2 \* 2**

**9 = 3 \* 3**

**16 = 4 \* 4**

**25 = 5 \* 5**

**---Izlazak iz funkcije f0---**

**Kraj programa**

* formatirajte program tako da ispisuje sljedeće poruke:

*Riješen C++ kôd se nalaze na stranici 136.*

*Pomoć:*

* ovdje ćete morati koristiti prototipove funkcija
* *for*-petlju izvršite u funkciji f0
* funkciji f0 moramo (iz main-a) reći koliko će puta ponoviti *for*-petlju: to možemo činiti na dva načina:

1. funkciji f0 ćemo proslijediti dva parametra za početak i kraj niza, a to su:
   * + aktuelni parametri u funkciji main: m i n
     + formalni parametri u funkciji f0: up i uk *(p za početak, k za kraj)*
2. funkciji f0 ćemo proslijediti samo jedan parametar za ukupan broj ponavljanja *for*-petlje, a to je:

* aktuelni parametar u funkciji main: vrijednost (n-m+1)
* formalni parametar u funkciji f0: ux

Funkciju f1 ćemo ostaviti neizmijenjenu!

**Zadatak 100:**

Slijedi zadatak sa ugniježđenom *for*-petljom u kojem nećete koristiti funkcije.

Koristeći izdvojeni dijagram toka za prost broj koji vam je dat u zadatku 100 (str. 95, workshop br. 9) napravite dijagram toka za sljedeći program:

Program treba izračunati sumu prostih brojeva od x do y. (Korisnik unosi vrijednosti x i y).

Dijagram toka riješite iz dva koraka:

* nacrtajte dio dijagrama za provjeru je li broj prost (*for*-petlja i *if-else* iskaza (dvostruki izbor)) - prepravite desni dio dijagrama toka iz zadatka br. 100
* nacrtajte (prepravite lijevi dio dijagrama toka iz zadatka br. 100) ostatak programa sa praznim prostorom u kojem bi se trebao nalaziti prethodno nacrtani dijagram toka

*Riješen dijagram toka i bez C++ kôda se nalaze na stranici 137.*

*Pomoć:*

* u lijevom dijagramu toka postavite početnu vrijednost varijable za sumu s na nulu
* pošto program treba samo računati sumu a ne ispisivati, morate u desnom dijagramu toka umjesto *if-else*-iskaza (dvostruki izbor) za ispis poruke '...je prost' ili '...NIJE prost' ubaciti *if*-iskaz (jednostruki izbor) koji će ako je broj n prost povećati sumu s za vrijednost n

**Zadatak 101:**

Napišite program (prepravite program iz prethodnih *workshop*-ova) koji će koristeći funkciju ispisi\_ako\_je\_prost ispisat proste brojeve iz datog raspona.

*Riješen C++ kôd se nalaze na stranici 137.*

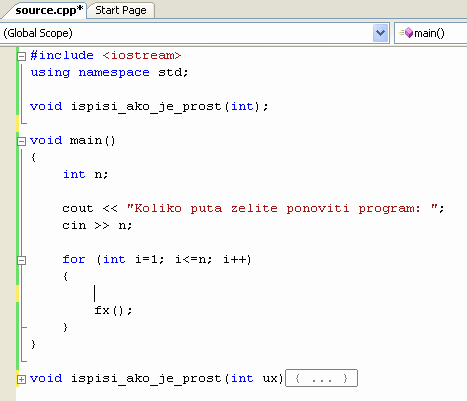
*Pomoć:*

* funkcija ispisi\_ako\_je\_prost treba provjeriti da li je ulazni parametar prost broj
* samo ako je ulazni parametar prost, funkcija treba ispisati taj broj
* funkcija ispisi\_ako\_je\_prost treba imati jedan ulazni parametar tipa *int*

**Zadatak 102:**

*Prepravljamo prethodni program:*

* ostavite neizmijenjen prototip i definiciju funkcije ispisi\_ako\_je\_prost
* u funkciji main unesite broj n na zahtjev korisnika (koliko puta želi da ponovi program)
* u *for*-petlju dodajte poziv nove funkcije fx() koja nema parametara
* izvršite *for*-petlju n puta

Preporučujemo, nakon što ste implementirali definiciju funkcije ispisi\_ako\_je\_prost i uspješno testirali u prethodnom zadatku, da sakrijete tijelo funkcije. Ako koristite Visual Studio 2003 ili 2005 ili 2008 to možete učiniti jednim klikom na znak '-' ispred funkcije:

**Visual Studio 2005**

Sakrivanjem kôda koji trenutno nećete mijenjati, lakše ćete moći održavati vaš program, a ujedno program postaje biti pregledniji.

Implementirajte funkciju fx, tako da ona poziva funkciju ispisi\_ako\_je\_prost za svaki cijeli broj iz niza [m,n]. Korisnik unosi brojeve m i n u funkciji fx.

Program formatirajte na sljedeći način.

*Riješen C++ kôd se nalaze na stranici 138.*

**Dijagram toka funkcije**

**Koliko puta zelite ponoviti program:** 3

=================1================

Unesi pocetak i kraj niza: 5 13

Broj 5 je prost.

Broj 7 je prost.

Broj 11 je prost.

Broj 13 je prost.

=================2================

Unesi pocetak i kraj niza: 5 9

Broj 5 je prost.

Broj 7 je prost.

=================3================

Unesi pocetak i kraj niza: 9 18

Broj 11 je prost.

Broj 13 je prost.

Broj 17 je prost.

Dijagram toka za program sa funkcijom se crta kao dva odvojena programa, s tim da se sadržaj funkcije main nalazi između **START** i **KRAJ**, a sadržaj pozvane funkcije (potprogram) se nalazi između '**PP**' (**p**očetak **p**rograma) i '**PUP**' (**p**ovratak **u** **p**rogram). Izraz za poziv funkcije (potprograma) se crta sa dvostrukim vertikalnim linijama.

|  |
| --- |
|  |

**Zadatak 103:**

Koristeći Visual Studio napravite neki koristan program koristeći DOS-ove naredbe u funkciji system. Evo primjera:

|  |  |
| --- | --- |
| 7:  8:  9:  10:  11:  12: | void formatiraj\_disketu()  {  cout << "Formatiranje usb memory stika: Pritisni 'Y' i ENTER \n";  system("format f: > null");  system("dir f:");  } |

***Primjer gotovih funkcija***

**Zadatak 104:**

Napraviti program koji će od korisnika zahtjevati unos pet brojeva (a,b,c,d,e). Potrebno je izračunati:

*  (uslov: *a≥0*)
*  (uslov: *b>0*)
*  (uslov: *c>0*)
*  (vrijednost *c* je u radijanima)
* 
*  (ako je *d=0*, onda *e* mora biti različito od *0,* jer je *00* nedefinisano)

*Riješen C++ kôd se nalaze na stranici 138.*

*Pomoć:*

|  |  |  |
| --- | --- | --- |
| Funkcija u matematici | Funkcija u C++ | Uslov |
|  | y = sqrt(x) | x≥0 |
|  | y = log10(x) | x>0 |
|  | y = log(x) | x>0 |
|  | y = cos(x) |  |
|  | y = sin(x) |  |
|  | y = pow(x1,x2) | (x1!=0) || (x2!=0)  barem jedna vrijednost mora biti različita od nule |

**Razlika između** void**-funkcije i funkcije sa povratnom vrijednošću**

Analizirajte pozive funkcije u sljedećem dijelu programa! Veoma je bitno da shvatite razlike između ove dvije vrste funkcija. Čitajte kôd pažljivo.

|  |  |
| --- | --- |
| 7:  8:  9:  10:  11:  12:  13:  14:  15:  16: | ...  void main()  {  float x;  **x = sqrt(9.0);** //poziv funkcije koja nije void  cout << "x = " << x << endl;  **ispisi\_ako\_je\_prost(17);** //poziv void-funkcije  }  ... |

Funkcija sqrt (linija br. 11):

* pozivana je funkcija koja **nije** tipa void, to znači da:
  + funkcija sqrt dodjeljuje vrijednost (broj 3) onoj varijabli kojoj je funkcija pridružena znakom jednakosti, tj. varijabli x
  + ova funkcija ne ispisuje nikakvu vrijednost na ekran

**Zadatak 105:**

Šta će sljedeći program ispisati:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23: | void main()  {  „sqrt(9.0)“ predstavlja vrijednost 3  „sqrt(4.0)“ predstavlja vrijednost 2  x = 3 + 2 \* 2  double x;    x = sqrt**(**9.0**)** + 2 \* sqrt**(**4.0**)**;  cout << "x = " << x << endl;  x = pow**(**sqrt(9.0)**,** 3**)**;  cout << "x = " << x << endl;  x = sqrt**(**sqrt(16.0)**)**;  cout << "x = " << x << endl;  x = pow**(**sqrt(sqrt(16.0))**,** sqrt(9.0)**)**;  cout << "x = " << x << endl;  if **(**sqrt(9.0) > sqrt(8.9)**)**  cout << sqrt(9.0) <<" je vece od " << sqrt(8.9) << endl;  } |

Funkcija ispisi\_ako\_je\_prost (linija br. 14):

* pozivana je funkcija koja **je** tipa void, to znači da:
  + funkcija ne vraća nikakvu vrijednost u program, već je izlaz ove funkcije poruka ispisana na ekran pomoću naredbe cout
  + ova funkcije se ne smije pridružiti (znakom jednakosti ili sl.) nekoj varijabli, niti se smije nalaziti u naredbi cout kao što to smije funkcija sqrt (linija br. 22)

*Pitanje:* Koja je povratna vrijednost funkcije system? Da li je void ili ne?

*Odgovor:* Funkcija system je istog tipa kao i funkcija ispisi\_ako\_je\_prost.

**Zadatak 106:**

Pronađite greške u pozivima funkcija u sljedećem programu uz pretpostavku da su prototipovi funkcija ispravni.

|  |  |
| --- | --- |
| 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: | #include <iostream>  #include <math.h>  using namespace std;  void f0(); // prototip funkcije  void f1(int); // prototip funkcije  void f2(int, int); // prototip funkcije  void f3(float); // prototip funkcije  void main()  {  double x = f1();  f0();  f2(3,4);  f2(sqrt(9.0), sqrt(4.0));  f2(pow(2.0), sqrt(4.0));  f3(pow(2.0, 2.0));  f1(f3());  pow(3,4);  x = sqrt(f1(), f3());  cout << f3() << endl;  cout << int (f3()) << endl;  char y = system("PAUSE");  }  ... |

*Rješenje se nalaze na stranici 139.*

**Funkcija sa povratnom vrijednošću *(funkcija koja nije void)***

Pogledajte ponovo funkciju volumen iz zadatka br. 120. Funkcija računa zapreminu u litrima za tri ulazna parametra koji predstavljaju dužinu, širinu i visinu bazena u metrima.

|  |  |
| --- | --- |
| 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 volumen(int u1, int u2, int u3)  {  int V;  V = u1 \* u2 \* u3 \* 1000; // kubni metar pomnožen sa 1000 = 1 litar!!  cout << "Zapremina: " << V << " m \n";  }  void main()  {  int a, b, c;  cout << "Unesi dimenzije u metrima:\n";  cin >> a >> b >> c;  volumen(a,b,c);  } |

U funkciji main unosimo vrijednosti za tri varijable (a, b, c) i prosljeđujemo ih funkciji volumen pri njenom pozivu. U funkciji volumen te tri varijable primamo kao u1, u2, u3. Funkcija volumen izračunava zapreminu V i ispisuje je na ekran. Varijabla V koja pamti zapreminu postoji samo u funkciji volumen a ne u funkciji main. Stoga funkcija main 'ne zna' vrijednost zapremine. Kad bi iz funkcije main htjeli npr. ispisati vrijednost zapremine na ekran ili u fajl, ne bi smo to mogli učiniti. Vrijednosti izračunate u ovakvim funkcijama, koje su tipa void (kao što je to funkcija volumen), mogu se ispisati samo na ekran a ne mogu se 'prenijeti' u funkciju main.

Prenos vrijednosti iz void-funkcije u funkciju main možemo postići kad bi koristili globalne varijable. U tom slučaju bi istu varijablu mogli koristiti u funkciji volumen za računanje zapremine i u funkciji main za neke druge operacije (kao što je ispis u fajl i ponovni ispis na ekran). Slijedi primjer sa globalnom varijablom.

|  |  |
| --- | --- |
| 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;  **int V;** //globalna varijabla  void volumen(int u1, int u2, int u3)  {  **V** = u1 \* u2 \* u3 \* 1000;  ~~cout << "Zapremina: " << V << " litara \n";~~  }  void main()  {  int a, b, c;  cout << "Unesi dimenzije u metrima:\n";  cin >> a >> b >> c;  volumen(a,b,c);  cout << "Zapremina u litrima iznosi: " << **V** << endl;  } |

Rekli smo već da nije preporučljivo koristiti globalne varijable, a u narednim zadacima slijedi objašnjenje. Najbolji način je koristiti funkcije sa povratnom vrijednošću (funkcije koje nisu void).

Za vježbu riješite sljedeći zadatak.

**Zadatak 107:**

Prepravite prethodni primjer tako da koristite funkciju sa povratnom vrijednošću (funkcija koja nije void) pod imenom vol sa tri ulazna parametra. Prepravite samo funkciju main. Kasnije ćete dodati prototip i definiciju funkcije vol.

Dodajte u funkciju main još jedan *if*-iskaz koji će, ako je zapremina veća od 1000, ispisati poruku „Bazen je velik!“.

Napravite dvije verzije:

* jednu sa korištenjem varijable V,
* drugu bez korištenja varijable V.

*Primjer prepravljene funkcije main se nalazi na stranici 139.*

**Prototip i definicija funkcije sa povratnom vrijednošću**

Možemo zaključiti, da ime funkcije vol predstavlja vrijednost zapremine koja je tipa *integer*. Znači da ova funkcija vol (koja je nije void) vraća vrijednost tipa *int*. Stoga ćemo u prototipu funkcije umjesto void koristiti int kao povratnu vrijednost.

*\*\*\* Pod povratnom vrijednošću se ne podrazumijeva vrijednost (zapremina) koja se ispisuje pomoću naredbe cout na ekran, već vrijednost (zapremina) koja se vraća iz neke funkcije u funkciju main.(Funkcija vraća vrijednost kroz svoje ime). \*\*\**

U prototipu funkcije vol govorimo da će funkcija vraćati vrijednost tipa *int*:

|  |  |
| --- | --- |
| 5: | int vol(int, int, int); // prototip |

Definicija funkcije koja nije void mora sadržavati naredbu koja će vraćati izračunatu vrijednost u funkciju koja je poziva, ovdje main. To je naredba return x; koja će vraćati vrijednost varijable x. Pogledajte liniju br. 9.

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9:  10: | int vol(int u1, int u2, int u3)  {  int x; // varijabla x bi trebala da bude tipa int, jer funkcija vraća int  x = u1 \* u2 \* u3 \* 1000;  return x; // vrijednost x se vraca u funkciju main  } |

Dodajte funkciju vol u prethodni program (zadatak br. 128).

**Zadatak 108:**

Napravite definicije za sljedeće prototipove funkcija:

|  |  |
| --- | --- |
| 5:  6:  7:  8:  9: | float sr\_vrij(float, float, float); // izlaz je aritetmicka srednina  int suma\_parnih\_brojeva(int, int); // izlaz je suma parnih brojeva od u1 do u2  float max(float, float); // izlaz je veci broj od u1 i u2  float hipotenuza(float, float); // izlaz je hipotenuza, a ulaz su dvije katete  float kateta(float, float); // izlaz je kateta, a ulaz je hipotenuza i kateta |

Dodajte u sljedeću nepotpunu funkciju main pozive gornjih funkcija.

|  |  |
| --- | --- |
| ...  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:  ... | ...  void main()  {  float a, b, c, y1;  cout << "Unesite tri broja \n";  cin >> a >> b >> c;  ...  cout << "Aritmeticka sredina od tri unesena broja je " << y1 << endl;  int m, n, y2;  cout << "\nUnesite pocetak i kraj niza \n";  cin >> m >> n;  ...  cout << "Suma parnih brojeva od " << m << " do " << n << " je " << y2 << endl;  float q, w, y3;  cout << "\nUnesite dva broja \n";  cin >> q >> w;  ...  cout << "Veci broj od dva unesena: " << y3 << endl;  float h, k1, k2;  cout << "\nUnesite dvije katete jednakostranicnog trougla \n";  cin >> k1 >> k2;  ...  cout << "Hipotenuza: " << h << endl;  float hi, ka1, ka2;  cout << "\nUnesite hipetnuzu i katetu \n";  cin >> hi >> ka1;  ...  cout << "Kateta: " << k2 << endl;  }  ... |

*Rješenje se nalazi na stranici 140.*

*Dodatak:* Slijede primjeri za pozive funkcija sa povratnom vrijednošću.

|  |  |
| --- | --- |
| 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: |  |

**Rješenja**

|  |  |
| --- | --- |
| Rješenje zadatka br. **99**: *(prvi način)* | |
| 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: | #include <iostream>  #include <math.h>  using namespace std;  void f1(int); // prototip (deklaracija) funkcije  void f0(int, int); // prototip (deklaracija) funkcije  void **main**()  {  int m, n;  cout << "Unesi pocetak i kraj niza: \n";  cin >> m >> n;  f0(m, n); //funkcija se poziva samo jednom  cout << "Kraj programa \n";  }  void **f0**(int up, int uk)  {  cout << "\n----Ulazak u funkciju f0---- \n";    for (int i=up; i<=uk; i++)  {  **f1**(i); // funkcije se poziva (n-m+1) odnosno (uk-up+1) puta  }  cout << "---Izlazak iz funkcije f0--- \n\n";  }  void **f1**(int u1)  {  ... // funkcije f1 je nepromijenjena |

|  |  |
| --- | --- |
| Rješenje zadatka br. **100**: *(drugi način)* | |
| 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: | #include <iostream>  #include <math.h>  using namespace std;  void f1(int); // prototip (deklaracija) funkcije  void f0(int); // prototip (deklaracija) funkcije  void main()  {  int m, n;  cout << "Unesi pocetak i kraj niza: \n";  cin >> m >> n;  f0(**n-m+1**); //funkciji se prosljeđuje se samo jedan parametar  cout << "Kraj programa \n";  }  void f0(int **ux**)  {  cout << "\n----Ulazak u funkciju f0---- \n";  for (int i=**1**; i<=**ux**; i++)  { // ostatak funkcije je isti kao u prvom načinu |

|  |
| --- |
| Rješenje zadatka br. **101**: |
|  |

|  |  |
| --- | --- |
| Rješenje zadatka br. **102**: | |
| 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: | #include <iostream>  using namespace std;  void ispisi\_ako\_je\_prost(int); // prototip  void **main**()  {  int x, y;    cout << "Unesite brojeve x i y \n";  cin >> x >> y;  cout << "===============\n";  for (int n=x; n<=y; n++)  {  ispisi\_ako\_je\_prost(n);  }  }  void **ispisi\_ako\_je\_prost**(int ux)  {  int b = 0;  for (int i=2; i<=ux-1; i++)  {  if (ux % i == 0)  {  b = b + 1;  }  }  if (b == 0)  cout << "Broj " << ux << " je prost. \n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **103**: | |
| 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: | #include <iostream>  using namespace std;  void ispisi\_ako\_je\_prost(int);  void fx();  void **main**()  {  int n;    cout << "Koliko puta zelite ponoviti program: ";  cin >> n;  for (int i=1; i<=n; i++)  {  cout << "\n\n================="<< i <<"================\n\n";  fx();  }  }  void **fx**()  {  int m, n;  cout << "Unesi pocetak i kraj niza: ";  cin >> m >> n;  for (int i=m; i<=n; i++)  {  ispisi\_ako\_je\_prost(i);  }  }  void **ispisi\_ako\_je\_prost**(int ux)  {  ... |

|  |  |
| --- | --- |
| Rješenje zadatka br. **104**: | |
| 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:  47:  48: | #include <iostream>  #include <math.h>  using namespace std;  void main()  {  float a, b, c, d, e, y1, y2, y3, y4, y5, y6;  cout << "Unesi broj a koji nije negativan: ";  cin >> a;    cout << "Unesi broj b koji je pozitivan: ";  cin >> b;  cout << "Unesi ugao c u radijanima (1 rad = 90 stepeni): ";  cin >> c;  cout << "Unesi bazu d: ";  cin >> d;  cout << "Unesi eksponent e: ";  cin >> e;  if (a>=0)  {  y1 = sqrt(a);  cout << "y1 = " << y1 << endl;  }  if (b > 0)  {  y2 = log10(b);  y3 = log(b);  cout << "y2 = " << y2 << endl;  cout << "y3 = " << y3 << endl;  }  y4 = cos(c);  y5 = sin(c);  cout << "y4 = " << y4 << endl;  cout << "y5 = " << y5 << endl;  if (d!=0 || e!=0)  {  y6 = pow(d,e);  cout << "y6 = " << y6 << endl;  }  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **105**: | |
| 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: | #include <iostream>  #include <math.h>  using namespace std;  void f0();  void f1(int);  void f2(int, int);  void f3(float);  void main()  {  double x = f1(); // f1 je tipa void  f0(); // OK  f2(3,4); // OK  f2(sqrt(9.0), sqrt(4.0)); // OK  f2(pow(2.0), sqrt(4.0)); // u funkciji pow se nalazi samo jedan parametar  f3(pow(2.0, 2.0)); // OK  f1(f3()); // f3 je tipa void  pow(3,4); // pow nije tipa void  x = sqrt(f1(), f3()); // f1 i f3 su tipa void  cout << f3() << endl; // f3 je tipa void  cout << int (f3()) << endl;// f3 je tipa void  char y = system("PAUSE"); // system je tipa void  }  ... |

|  |  |
| --- | --- |
| Rješenje zadatka br. **106** - prvi način – sa varijablom V: | |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14: | void main()  {  int a, b, c, V; // varijabla V je sada lokalna  cout << "Unesi dimenzije u metrima:\n";  cin >> a >> b >> c;  V = vol(a,b,c);  cout << "Zapremina u litrima iznosi: " << **V** << endl;  if (**V** > 1000)  cout << "Bazen je velik! \n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **107** - drugi način – bez varijable V: | |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12: | void main()  {  int a, b, c;  cout << "Unesi dimenzije u metrima:\n";  cin >> a >> b >> c;  cout << "Zapremina u litrima iznosi: " << **vol(a,b,c)** << endl;  if **(** **vol**(**a,b,c**) > 1000 **)**  cout << "Bazen je velik! \n";  } |

|  |  |
| --- | --- |
| Rješenje zadatka br. **108**: | |
| ...  44:  45:  46:  47:  48:  49:  50:  51:  52:  53:  54:  55:  56:  57:  58:  59:  60:  61:  62:  63:  64:  65:  66:  67:  68:  69:  70:  71:  72:  73:  74:  75:  76:  77:  78:  79:  80: | ...  float sr\_vrij(float u1, float u2, float u3)  {  float x;  x = (u1 + u2 + u3)/3;  return x;  }  int suma\_parnih\_brojeva(int u1, int u2)  {  int s = 0;  for (int i=u1; i<=u2; i++)  {  if (i%2 == 0)  s = s + i;  }  return s;  }  float max(float u1, float u2)  {  if (u1 > u2)  return u1; // mozemo koristiti i vise return naredbi  else  return u2;  }  float hipotenuza(float uk1, float uk2)  {  return sqrt(uk1\*uk1 + uk2\*uk2);  } // u naredbi return mogu se nalaziti matematicki izrazi  float kateta(float u1, float u2)  {  if (u1 > u2) // hipotenuza je veca od katete  return sqrt(u1\*u1 - u2\*u2);  else  return sqrt(u2\*u2 - u1\*u1);  } |