1. Zapiši program, ki izračuna vrednosti funkcije za liha števila med 10 in 120.  
Uporabi ***for*** zanko.

Izračunane vrednosti funkcije naj v obliki **y = "rezultat"** sproti izpisuje, vsak rezultat v novo vrstico (*rezultat je izračunana vrednost*). 10 t

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

float y;

//za liha stevila med 10 in 120

for(int x =11; x<=119; x = x+2)//vsako drugo je liho

{

y = 2\*(pow(x,3) - 5\*sqrt(x));

cout<<"f("<<x<<") = "<<y<<endl;

}

return 0;

};

2. Zapiši program, ki bo računal obsege kroga toliko časa, dokler je vneseni polmer večji   
od 0. Uporabi ***do-while*** zanko ().

Znotraj zanke naj sproti izpisuje izračunane ploščine v obliki **P = \***, kjer je namesto \* izpisan izračunan obseg kroga. Vsak izpis v naj bo svoji vrstici.

Namesto števila uporabi konstanto z vrednostjo Pi = 3.14159. 10 t

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

float r, ob;

const float Pi = 3.14159;

//do - while: prvo branje pred zanko, v zanki na koncu

cout<<"Vnesi polmer: ";

cin>>r;

do

{

ob = 2\*Pi\*r;

cout<<"r = "<<r<<"; o = "<<ob <<endl;

cout<<"Vnesi polmer: ";

cin>>r;//pri r = 0, konča takoj

}while(r>0);

return 0;

};

3. Z uporabo ***while*** zanke, zapiši program, ki bo sešteval vnesena pozitivna cela števila toliko časa,

dokler je vsota seštetih števil manjša od 4000.

Sešteva naj le števila, ki so manjša od 50.

Če je število večje ali enako 50 naj izpiše: "**Število je preveliko.".**

Na koncu, zunaj zanke, naj program naj izpiše, z ustreznim komentarjem, izračunano vsoto. 15 t

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

unsigned int vsota = 0, st;

while(vsota<4000)

{

cout<<"vnesi stevilo: ";

cin>>st;

if(st<50)

vsota = vsota + st;//vsota += st;

else

cout<<"Stevilo je preveliko.\n";

cout<<"vsota sestetih stevil je "<<vsota<<".\n";

};

return 0;

};

int myNum = 5;               // Integer (whole number without decimals)  
double myFloatNum = 5.99;    // Floating point number (with decimals)  
char myLetter = 'D';         // Character  
string myText = "Hello";     // String (text)  
bool myBoolean = true;       // Boolean (true or false)

**const** int myNum = 15;  // myNum will always be 15  
myNum = 10;  // error: assignment of read-only variable 'myNum'

array / tabele

int main() {

string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};

for(int i = 0; i < 4; i++) {

cout << cars[i] << "\n";

}

return 0;

}

tabelaBranjeFunkcija

int main()

{

const int N = 70; //med -5 in 10 velikost tabele

float VredF[N] = {};

//potrebujemo dva stevca: x, i - mesta v tabeli

//ker sta razlicnih tipov, jih deklariramo pred for zanko

int i; //deklariramo lahko kjerkoli, a pred uporabo

float x; //nepisano pravilo: na zacetku programa

for (i = 0, x = -5; x <= 10; i++, x = x + 0.25) //preverjamo le x

{

VredF[i] = 2 \* pow(x, 3) - 2 \* pow(x, 2) + 3 \* x + 4;

cout << "VredF[" << i << "] za x = " << x << " je " << VredF[i] << endl;

}

// z While zanko

cout << "\nZ While zanko:\n";

i = 0;

x = -5;

while (x <= 20) //ce zelimo izracunati le 20 vrednosti

{

VredF[i] = 2 \* pow(x, 3) - 2 \* pow(x, 2) + 3 \* x + 4;

cout << "VredF[" << i << "] za x = " << x << " je " << VredF[i] << endl;

i++;

x = x + 0.25;

}

return 0;

};

MIN – MAX

int main()

{

const int N = 100;

int vnosi = 0; //steje vnose

float StA[N] = {0}, StB[N] = {}, x;

float vsota = 0;

float povprecje = 0, minimum, maksimum; //domaca naloga najmanj 15 stevil

cout << "-------------------------------" << endl;

cout << "Vnos stevil konca tako da pritisnes katerokoli tipko s crko" << endl;

cout << "-------------------------------" << endl;

cout << "x = ";

cin >> x;

while (cin)

{

StA[vnosi] = x; //shranimo podatek prebran pred zanko

vnosi++;

cout << "x = ";

cin >> x;

}

//izpis v obratnem vrstnem redu

for (int i = vnosi - 1; i >= 0; i--) //vnosi-1: ker mesta v tabeli zacnemo z 0

{

cout << "StA[" << i << "] => " << StA[i] << endl;

}

/\*maksimum = INT\_MIN; //najmanjse

minimum = INT\_MAX; //najvecje

cout << "Najvecje celo " << INT\_MAX << ", najmanjse celo " << INT\_MIN << endl;\*/

//izpis negativnih

cout << "-------------------------------" << endl;

cout << "Izpis negativnih: " << endl;

for (int i = 0; i < vnosi; i++)

{

if (StA[i] < 0)

{

cout << StA[i] << endl;

}

}

maksimum = StA[0];

minimum = StA[0];

for (int i = 0; i < vnosi; i++)

{

if (maksimum < StA[i])

{

maksimum = StA[i];

}

if (minimum > StA[i])

{

minimum = StA[i];

}

vsota += StA[i];

}

povprecje = vsota / vnosi;

cout << "-------------------------------" << endl;

cout << "Povprecje vnesenih stevil je: " << povprecje << endl;

cout << "-------------------------------" << endl;

cout << "Najvecjo stevilo je: " << maksimum << endl;

cout << "-------------------------------" << endl;

cout << "Najmanjse stevilo je: " << minimum << endl;

cout << "-------------------------------" << endl;

return 0;

};

/\*

Deklarija tabelo, ki vsebuje pozitivna cela stevila. Velikost in vrednost v njej so razvidne iz skice: 3,5,4,<siva>1<siva>,2,3,4,5, , .

Zapisi program, ki bo v tabelo vrinil stevilko 9 v celico, obarvano z sivo barvo.

Ostala stevila pa v istem vrstem redu pomakne za eno mesto desno.

\*/

#include <iostream>

#include <string>

using namespace std;

int main()

{

int N = 10; //velikost tabele

//tip ImeTabele[velikost] = {vrednosti...}

unsigned int PozSt[N] = {3, 5, 3, 1, 2, 3, 4, 5};

//izpis vnesenih stevil

for (int i = 0; i < N; i++)

{

cout << PozSt[i] << ' ';

}

cout << endl;

//premik stevil od sivega polja naprej za 1 desno

//da ne "povozimo" ze vneseni vrednosti, zacnemo na desni

for (int i = 8; i >= 4; i--)

{

PozSt[i] = PozSt[i - 1]; //vrednost na desni dobi vrednost predhodnika

//na 3. mesto (steti zacnemo z 0) vstavimo vrednost 9

}

PozSt[4] = 9;

for (int i = 0; i < N; i++)

{

cout << PozSt[i] << ' ';

}

cout << endl;

return 0;

};

/\*

pretvori vneseno desetisko stevil v dvojisko

\*/

#include <iostream>

using namespace std;

int main()

{

long long celidel, desetisko; //celidel uporabljamo v postopku----to stevilo vnesemo

long long dvojisko = 1; //zacetna vrednost dvojiskega

long long ostanek; //tu shranimo vneseno stevilo

cout << "Vnesi desetisko stevilo: ";

cin >> desetisko;

celidel = desetisko; //shranimo prebrano stevilo, ker ga v postopku spreminjamo

while (celidel > 0)

{

ostanek = celidel % 2; //ostanek, desni stolpec

dvojisko = dvojisko \* 10 + ostanek;

celidel = celidel / 10; //123/10 = 12 le celi del, ker deilomo celo stevilo s celim

};

cout << desetisko << " Dvojisko: " << dvojisko << endl;

return 0;

};

/\*iz dvojiskega v desetisko st\*/

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

unsigned long i, n, stevilo = 0, d;

cout << "Vnesi desetisko stevilo: ";

cin >> n;

cout << "Decimalna pretvorba " << n << " je ";

for (i = 0; n != 0; ++i)

{

d = n % 10;

stevilo = (d) \* (pow(2, i)) + stevilo;

n = n / 10;

}

cout << stevilo;

return 0;

}

/\* Zapiši program, ki izračuna vrednosti funkcije f(x)= 3x^3-2x

za soda števila med 11 in 111.

Izračuna in izpiše naj povprečno vrednost izračunanih vrednosti.

Uporabi for zanko.\*/

int main()

{

float x, y, povpr=0;

int n = 0;

//soda med 11 in 111

for(float x = 12; x<=110; x=x+2)

{

y = 3\*pow(x,3)-2\*x;

povpr = povpr+y;//vsoto shranimo kar v povpr

n++;//preštejemo, koliko števil

};

povpr = povpr/n;

cout<<endl<<endl<<"povpr = "<<povpr<<endl;

return 0;

};