MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE NATIONAL TECHNICAL UNIVERSITY

“KHARKIV POLYTECHNIC INSTITUTE”

DEPARTMENT OF SOFTWARE ENGINEERING AND MANAGEMENT INFORMATION TECHNOLOGIES

### PROGRAMING BASICS

### Laboratory Training 2

# C++ Operators and Statements

Executed by: SALEM CHAOUFI of group КН-221 iв.e

Teacher : olexiy Kizilov

KHARKIV 2021

## Training Tasks

### 1 Programmatic Implementation of Branching Algorithm

Develop a program that implements an algorithm for solving quadratic equation. The program should include checking all possible variants of the source data. In particular, the discriminant should be checked, and it should be checked whether the equation is quadratic. If the equation degenerates into a linear one, it is necessary to provide for finding the root of this linear equation, or to establish the presence of infinite count of solutions (absence of solutions).

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main() {

double a, b, c;

cout << "input a : ";

cin >> a;

cout << "input b : ";

cin >> b;

cout << "input c : ";

cin >> c;

double d, x1, x2, x;

d = pow(b, 2) - (4 \* a \* c);

if (a != 0 && b != 0 && c != 0 && d == 0) {

x = (-b) / (2 \* a);

cout << "the result is : " << x << endl;

}

else if (a != 0 && b != 0 && c != 0 && d > 0) {

x1 = ((-b - sqrt(d)) / (2 \* a));

x2 = ((-b + sqrt(d)) / (2 \* a));

cout << "the result is : x1= " << x1 << " x2= " << x2 << endl;

}

else if (a != 0 && b != 0 && c != 0 && d < 0) {

cout << "no roots" << endl;

}

else if (a == 0 && b != 0 && c != 0) {

x = (-c) / b;

cout << "the result is : x= " << x << endl;

}

else if (a == 0 && b == 0 && c != 0) {

cout << "NO ROOTS" << endl;

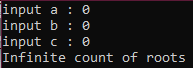
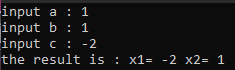
}

else if (a == 0 && b == 0 && c == 0) {

cout << "Infinite count of roots" << endl;

}

}



### 2 Programmatic Implementation of Looping Algorithm

Develop a program that implements an algorithm for calculating the following expression:

*y* = 1/(*x* + 2) + 2/(*x* + 4) + ... + (*k* - 1)/(*x* + 2(*k* - 1)) + (*k* + 1)/(*x* + 2(*k* + 1)) + ... + *n*/(*x* + 2*n*)

Provide a check of possible errors.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main()

{

int x, n, k;

int i = 1;

cout << "input x, n, k" << endl;

cin >> x >> n >> k;

double y = 0;

while (i <= n && x + (2 \* n) != 0) {

for (i = 1; i <= k - 1; i++) {

if (x + (2 \* i) == 0) {

cout << "i =" << i << endl << "x + (2 \* i) == 0,error";

return 0;

}

y += (double)i / (x + (2 \* i));

}

for (i = k + 1; i <= n; i++) {

if (x + (2 \* i) == 0) {

cout << "i =" << i << endl << "x + (2 \* i) == 0,error";

return 0;

}

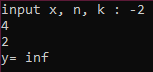
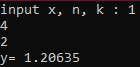
y += (double)i / (x + (2 \* i));

}

}

cout << "y= " << y;

}



### 3 Calculating Product

Write a program that reads **x** and **n** and calculates **y**:

*y* = (*x* + 1)(*x* - 2)(*x* + 3)(*x* - 4) ... (*x* - 2*n*)

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main(){

double x , n ;

double y = 1 ;

cout << "input x : " ;

cin >> x ;

cout << "input n : " ;

cin >> n ;

for (int i = 1 ; i <= 2\*n ; i++){

if(i % 2 == 0){

y \*= x - i ;

}

else {

y \*= x + i ;

}

}

cout << "the result is : " << y << endl ;

}



### 4 Calculating Sum

Write a program that reads **eps** and calculates **y**:

*y =* 1/2 + 1/4 + 1/8 + 1/16 + ...

The loop terminates if new summand is less than **eps**.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main()

{

double eps ;

double y = 0 ;

double eps1 = 0.5 ;

int count = 1 ;

cout << "Enter eps = " ;

cin >> eps ;

while (eps1 > eps){

y += eps1 ;

eps1 /= 2 ;

}

cout << "Y = " << y << endl ;

}



### 5 Individual Assignment

You should develop a program that calculates values of a function in a given range. The program should implement an algorithm developed in carrying out assignment 1.3 of [previous lab](http://iwanoff.inf.ua/programming_1_en/LabTraining01.html).

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main() {

int x, n, a, b;

double p = 1, s = 0;

cout << "Input n>=1, n= ";

cin >> n;

cout << "Input x= ";

cin >> x;

if (x < 0)

{

for (int i = 1; i <= n; i++)

{

s += (double) pow((i + x), 2);

}

cout << "x= " << x << " s= " << s << endl;

s = 0;

}

else

{

for (double i = 0; i <= n - 1; i++)

{

for (double j = 1; j <= n; j++)

{

p \*=(double) (x + i) / (i + j);

}

s = s + p;

p = 1;

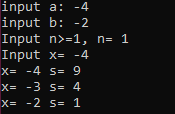
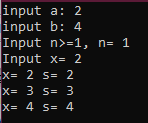
}

cout << "x= " << x << " s= " << s << endl;

s = 0;

}

}



CONCLUSION :

Because of this part, now I know how to code an algorithm using c++.