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 3

# Use of Functions

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

Teacher : olexiy Kizilov

KHARKIV 2021

## 1 Training Tasks

### 1.1 Static Local Variables

Write a program that calculates and shows the minimum and maximum of integers as the user inputs those integers. Use static local variables.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

void maxmin() {

int i, n, num;

cout << "input how much numbers you want : ";

cin >> n;

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

cout << "input number " << i << " : " << endl;

cin >> num;

static int min;

static int max;

if (i == 1) {

max = num;

min = num;

}

if (max < num) {

max = num;

}

if (min > num) {

min = num;

}

if (i == n) {

cout << "max is : " << max << " , min is : " << min << endl;

}

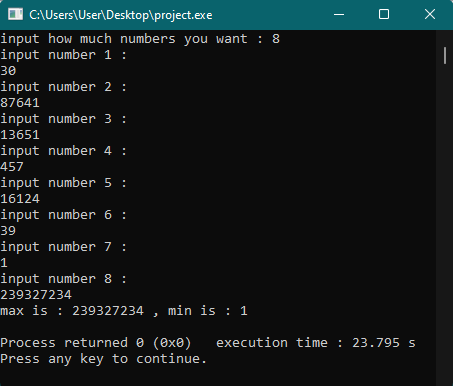
}

}

int main() {

maxmin();

}



### 1.2 Recursion

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

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

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int recurse(int n, int x){

if(n > 0){

return (n + x) \* recurse(n-1, x);

}

else{

return 1;

}

}

int main() {

int x , n ;

cout << "input x: " ;

cin >> x ;

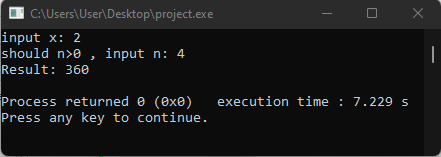
cout << "should n>0 , input n: " ;

cin >> n ;

int y = recurse(n, x);

cout << "Result: " << y <<endl ;

}



### 1.3 Default Arguments

Create a function that returns 1, argument, and product of arguments, depending on arguments count. Test this function in main() function. Implement program in two ways: using function overloading and using default arguments.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int func() {

return 1 ;

}

int func(int x) {

return x ;

}

int func(int x , int y) {

return x \* y ;

}

int func(int x , int y , int z) {

return x \* y \* z ;

}

int main() {

cout << func() << endl ;

cout << func(10) << endl ;

cout << func(4 , 5) << endl ;

cout << func(1 , 1 , 1)<< endl;

return 0 ;

}



### 1.4 Quadratic Equation

Create a function for solving quadratic equations. The function should return the number of roots (0, 1, or 2) or -1 if the equation has an infinite number of roots. The function should get the coefficients as arguments and return the roots as reference-type arguments.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

double RootsF() {

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;

}

return 0.1;

}

int main() {

return RootsF();

}

### 

### 1.5 Individual Assignment

You should create a program that implements an individual assignment of [previous laboratory training](http://iwanoff.inf.ua/programming_1_en/LabTraining02.html). Program should be split into several functions. Function y() should obtain values of x and n as arguments and return value calculated using formula given in an individual assignment. Create a separate function for reading data. Do not use global variables.

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

double y(int x, int n)

{

double s = 0;

double p = 1;

if (x < 0)

{

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

{

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

}

return s;

s = 0;

}

else

{

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

{

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

{

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

}

s = s + p;

p = 1;

}

return s;

s = 0;

}

}

int main()

{

double n, a, b;

double p, s;

cout << "input a: ";

cin >> a;

cout << "input b: ";

cin >> b;

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

cin >> n;

if (a <= b)

{

for (int x = a; x <= b; x++)

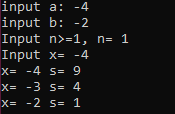
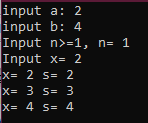
{

cout << "x= " << x << " y= " << y(x, n) << endl;

}

}

}



CONCLUSION :

Because of this lab, now I know how to deal with functions of all kinds in c++.