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

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 test(int i , int n , int 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() {

double num , n ;

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

cin >> n ;

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

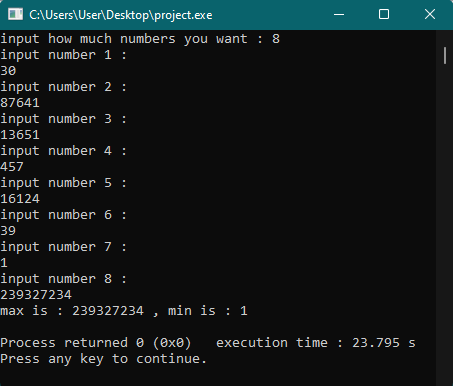
cout << "input number " << i << " : " << endl ;

cin >> num ;

test( i , n , num) ;

}

}



### 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 ;

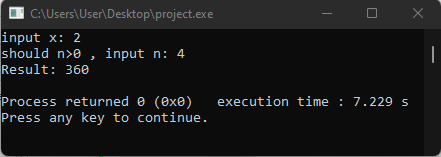
cout << "should n>0 , input n: " ;

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;

int roots( double a , double b , double c ){

double d , x1 , x2 , x;

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

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

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

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

}

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

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 ;

}

else {

cout << "no solution" << endl ;

}

}

int main() {

double a , b , c ;

double d , x1 , x2 , x;

cout << "input a : " ;

cin >> a ;

cout << "input b : " ;

cin >> b ;

cout << "input c : " ;

cin >> c ;

return roots(a , b , c) ;

}

### 

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

{

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

}

return s;

s = 0;

}

else

{

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

{

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

{

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

}

s = s + p;

p = 1;

}

return s;

s = 0;

}

}

int main()

{

int 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++)

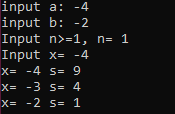
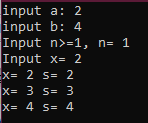
{

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

}

}

}



## 2 - Exercises

1. Create a program that tests signum() function :

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int signum(double x) {

return (x < 0) ? -1 : ((x > 0) ? 1 : 0);

}

int main() {

double x ;

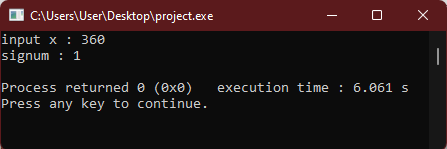
cout << "input x : " ;

cin >> x ;

cout << "signum : " << signum(x) << endl ;

return 0 ;

}



2. Develop and test a function that calculates product of three arguments

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int calc(double a , double b , double c){

return a\*b\*c ;

return 0 ;

}

int main(){

double a , b , c ;

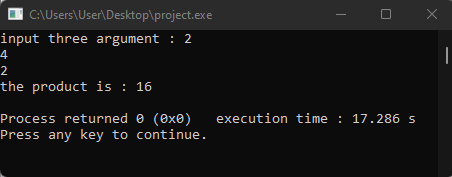
cout << "input three argument : " ;

cin >> a >> b >> c ;

cout << "the product is : " << calc(a,b,c) << endl ;

return 0 ;

}



3. Develop and test a function that calculates product of n first odd values

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int calc (int n ){

int result = 1 ;

int j = 1 ;

while (n > 0){

if (j % 2 != 0){

result \*= j ;

n-- ;

}

j++;

}

return result ;

}

int main (){

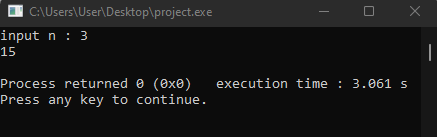
int n ;

cout << "input n : " ;

cin >> n ;

cout << "result is : " << calc(n) << endl ;

}



5. Develop and test a function that calculates factorial

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int factorial(int n ,int s = 1 ){

for (int i = n ; i >= 1 ; i--){

s \*= i ;

}

return s ;

}

int main() {

int n ;

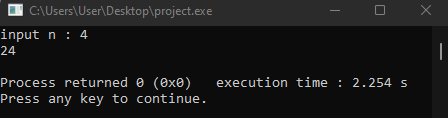
int s = 1 ;

cout << "input n : " ;

cin >> n ;

cout << factorial(n,s) << endl ;

}



6. Develop and test a function that prints all even values in a given range

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

void calc (int a , int b){

for (int i = a ; i <= b ; i++){

if( i % 2 == 0 )

cout << i << endl ;

}

}

int main() {

int a , b ;

cout << "should a<b input a and b : " << endl ;

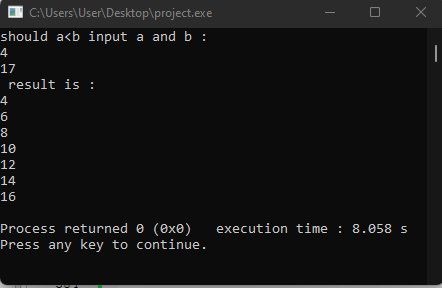
cin >> a ;

cin >> b ;

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

calc(a,b) ;

}



7. Develop and test a function and prints product of n first even values

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int calc (int n ){

int result = 1 ;

int j = 1 ;

while (n > 0){

if (j % 2 == 0){

result \*= j ;

n-- ;

}

j++;

}

return result ;

}

int main (){

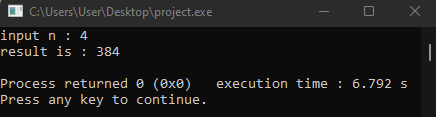
int n ;

cout << "input n : " ;

cin >> n ;

cout << "result is : " << calc(n) << endl ;

}



8. Develop and test a function that calculates greatest common divisor of two integers

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int gcd(int a, int b)

{

if (a == 0)

return b;

if (b == 0)

return a;

if (a == b)

return a; // return any one of them

if (a > b)

return gcd(a-b, b);

return gcd(a, b-a);

}

int main()

{

int a , b ;

cout << "input a & b : " ;

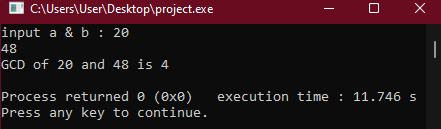
cin >> a ;

cin >> b ;

cout<<"GCD of "<<a<<" and "<<b<<" is "<<gcd(a, b) << endl ;

return 0;

}



CONCLUSION :

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