МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 10**

по дисциплине: ”Основы алгоритмизации и программирования”

на тему: ***”Указатели и функции”***

Вариант 3

Выполнил**:** студент группы *10701219* Харлап С. А.

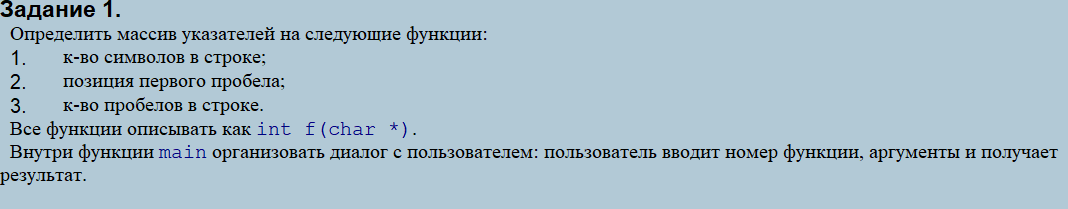
Приняла**:** ст. пр. Борисова И. М.

Минск 2020

# Лабораторная работа № 10

Цель работы: Научиться использовать указатели при работе с функциями.

## Задание 1.



### Код программы

// Лабораторная работа № 10. Указатель на функцию

// Задание 1

// Вариант 3.

// Студента группы 10701219 Харлапа Сергея Александровича

#include <iostream>

using namespace std;

int Len(char\* string) {

int len = 0;

for (int i = 0; string[i] != '\0'; i++) {

len++;

}

return len;

}

int FirstSpaseID(char\* string) {

int pos = -1;

for (int i = 0; string[i] != '\0'; i++) {

if (string[i] == ' ') {

pos = i;

break;

}

}

return pos;

}

int SpaceAmount(char\* string) {

int counter = -0;

for (int i = 0; string[i] != '\0'; i++) {

if (string[i] == ' ') {

counter++;

}

}

return counter;

}

void DoFunction(int(\*functions)(char\*)){

char string[100];

getchar();

cout << "enter string ";

cin.getline(string, 100);

cout << functions(string) << endl;

}

int main()

{

int (\*functions[3])(char\* );

functions[0] = Len;

functions[1] = FirstSpaseID;

functions[2] = SpaceAmount;

while (true) {

int chooser;

bool breakFlag = false;

cout << "enter 0 for getstringline\nenter 1 for finde first space\nenter 2 for find amount spaces\n\

else programm will stoped ";

cin >> chooser;

switch (chooser) {

case 0: {

DoFunction(functions[0]);

break;

}

case 1: {

DoFunction(functions[1]);

break;

}

case 2: {

DoFunction(functions[2]);

break;

}

default: breakFlag = true;

}

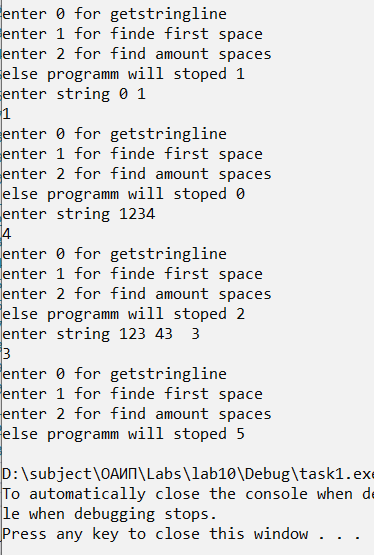
if (breakFlag) {

break;

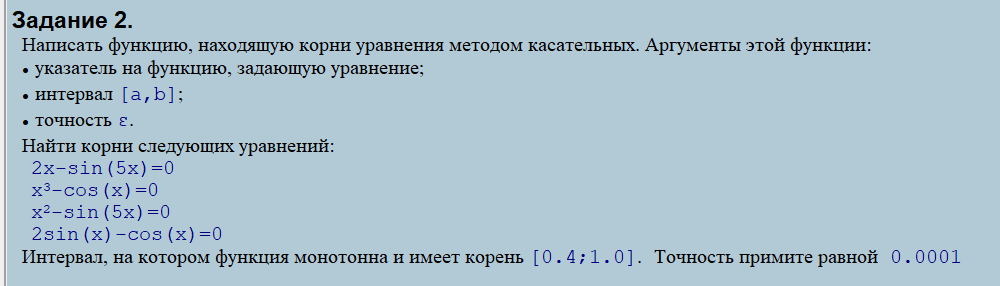
}

}

### Скриншоты результатов



## Задание 2.



### Код программы

// Лабораторная работа № 10. Указатель на функцию

// Задание 2

// Вариант 3.

// Студента группы 10701219 Харлапа Сергея Александровича

#include <iostream>

#include <cmath>

using namespace std;

double F1(double x) {

double f\_x = 2 \* x - sin(5 \* x);

double f\_\_x = 2 - 5 \* cos(5 \* x);

return f\_x / f\_\_x;

}

double F2(double x) {

double f\_x = x\*x - cos(x);

double f\_\_x = 2\*x + sin(x);

return f\_x / f\_\_x;

}

double F3(double x) {

double f\_x = x \* x - sin(5 \* x);

double f\_\_x = 2 \* x - 5 \* cos(5 \* x);

return f\_x / f\_\_x;

}

double F4(double x) {

double f\_x = 2 \* sin(x) - cos(x);

double f\_\_x = 2 \* cos(x) + sin(x);

return f\_x / f\_\_x;

}

double const eps = 0.0001;

double CalculateFunctionByConsernLineMethod(double a, double b, double x, double (\*functionPatter)(double)) {

double df = functionPatter(x);

if (abs(df) < eps) return x;

return CalculateFunctionByConsernLineMethod(a, b, x - df, functionPatter);

}

int main() {

cout << "F1 " << CalculateFunctionByConsernLineMethod(0.4, 1, 0.4, F1) << endl;

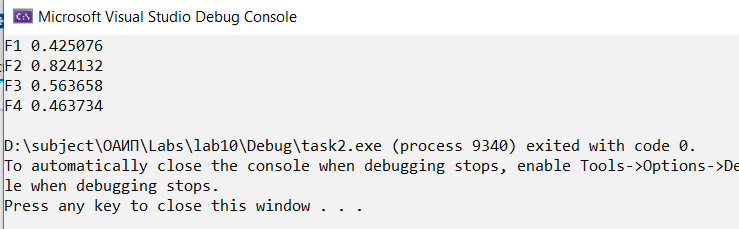
cout << "F2 " << CalculateFunctionByConsernLineMethod(0.4, 1, 0.4, F2) << endl;

cout << "F3 " << CalculateFunctionByConsernLineMethod(0.4, 1, 0.4, F3) << endl;

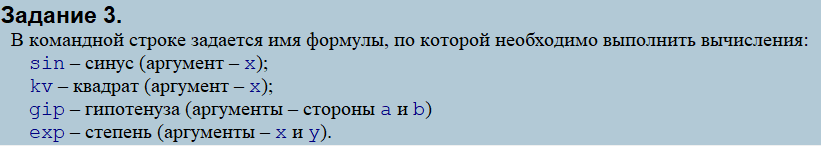
cout << "F4 " << CalculateFunctionByConsernLineMethod(0.4, 1, 0.4, F4) << endl;

return 0;}

Скриншоты результатов



## Задание 3.



### Код программы

// Лабораторная работа № 10. Указатель на функцию

// Задание 3

// Вариант 3.

// Студента группы 10701219 Харлапа Сергея Александровича

#include <iostream>

#include <cmath>

using namespace std;

double Sin(double x) {

return sin(x);

}

double (Kv)(double x) {

return x \* x;

}

double (Gip)(double a, double b) {

return sqrt(Kv(a) + Kv(b));

}

double (EXP)(double x, double y) {

double ans = 1;

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

ans \*= x;

}

return ans;

}

double Parser(char\* boof) {

double integer = 0;

double remainder = 0;

double divider = 10;

bool pointFlag = false();

int i = 0;

while (('0' <= boof[i] && boof[i] <= '9') || boof[i] == '.') {

if (boof[i] == '.') {

pointFlag = true;

i++;

continue;

}

if (!pointFlag) {

integer \*= 10;

integer += int(boof[i] - '0');

}

else {

remainder += int(boof[i] - '0');

remainder \*= 10;

divider \*= 10;

}

i++;

}

return integer + remainder/divider;

}

int main(int argc, char \*\*argv)

{

double (\*pToSin)(double) = Sin;

double (\*pToKv)(double) = Kv;

double (\*pToGip)(double, double) = Gip;

double (\*pToExp)(double, double) = EXP;

while (true) {

char\* boof = new char[10];

boof = argv[1];

if (strcmp(boof, "sin") == 0) {

double x = Parser(argv[2]);

cout << "sin(x) - " << pToSin(x);

}

else if (strcmp(boof, "kv") == 0) {

double x = Parser(argv[2]);

cout << "x^2 - " << pToKv(x);

}

else if(strcmp(boof, "gip") == 0) {

double a = Parser(argv[2]), b = Parser(argv[3]);

cout << "gip(a,b) - " << pToGip(a, b);

}

else if (strcmp(boof, "exp") == 0) {

double x = Parser(argv[2]), y = Parser(argv[3]);

cout << "x^y - " << pToExp(x, y);

}

else {

cout << "incorrect";

}

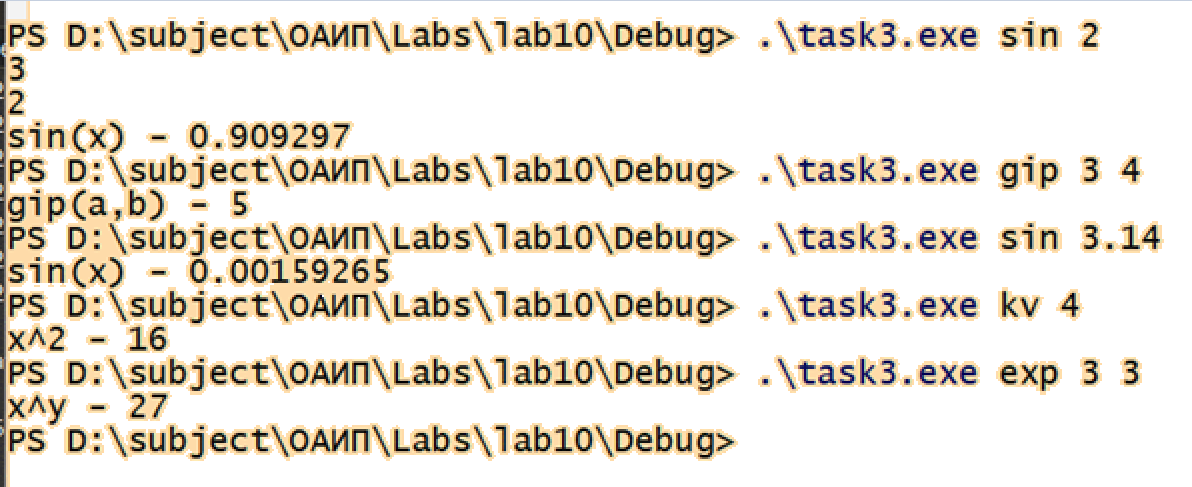
break;

}

return 0;

### `

### Скриншоты результатов



## Вывод:

Научились использовать указатели при работе с функциями