**Ф****ЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ПРАКТИКУМУ

по дисциплине

**ПРОГРАММИРОВАНИЕ В СИСТЕМАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ**

**Программирование на языке C**

Выполнил студент группы БИН1705:

Нагорный И.А.

Москва 2019

# Практикум №1 Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Линейные алгоритмы

### Цель практикума

Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки и научится создавать линейные программы на языке C с применением арифметических операций.

### Задание

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

### Индивидуальный вариант задания

### Решение заданий

#### Задание 1

#include <stdio.h>

#include <math.h>

//Вариант 40

float func(float x){

float f = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

return f;

}

void main(){

float x = 5;

float fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

printf("\nEnter x:\n");

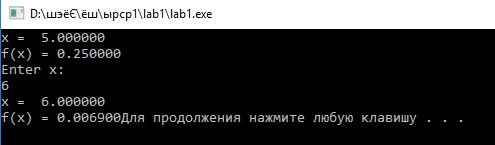
scanf("%f", &x);

fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

system ("pause");

}



#### Задание 2

#include <stdio.h>

#include <math.h>

float func(float x){

float f = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

return f;

}

void main(){

float x = 5;

float fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

printf("\nEnter x:\n");

scanf("%f", &x);

fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

system("pause");

}

Результат аналогичен.

#### Задание 3

#include <stdio.h>

#include <math.h>

float func(float x);

void main(){

float x = 5;

float fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

printf("\nEnter x:\n");

scanf("%f", &x);

fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

}

float func(float x){

float f = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

return f;

}

Результат аналогичен.

#### Задание 4

#include <stdio.h>

#include <math.h>

float x, result;

void main(){

x=5;

func();

printf("x = %f\nf(x) = %f", x, result);

printf("\nEnter x:\n");

scanf("%f", &x);

func();

printf("x = %f\nf(x) = %f", x, result);

}

void func(){

result = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

}

Результат аналогичен.

#### Задание 5

Функция task5\_func.c

#include <math.h>

float func(float x){

float f = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

return f;

}

Task5\_main.c

#include <stdio.h>

#include <math.h>

#include "task5\_func.c"

//Вариант 40

void main(){

float x = 5;

float fx = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

printf("x = %f\nf(x) = %f", x, fx);

printf("\nEnter x:\n");

scanf("%f", &x);

fx = func(x);

printf("x = %f\nf(x) = %f", x, fx);

getchar();

}

Результат аналогичен.

#### Задание 6

Task6\_func.c

#include <math.h>

float x, result;

void func(){

result = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

}

Task6\_main.c

#include <stdio.h>

#include "task6\_func.c"

//Âàðàèíò 40

extern float x, result;

extern void func();

void main(){

x=5;

func();

printf("x = %f\nf(x) = %f", x, result);

printf("\nEnter x:\n");

scanf("%f", &x);

func();

printf("x = %f\nf(x) = %f", x, result);

}

*Task6\_main..c*

Результат аналогичен.

#### Задание 7

func.h

#ifndef FUNC\_H\_INCLUDED

#define FUNC\_H\_INCLUDED

double x, result;

void f();

#endif // FUNC\_H\_INCLUDED

Task7\_func.c

#include <math.h>

#include "func.h"

extern double x, result;

extern void f(){

result = (float) 1/4-1/4\*sin(5/2\*3.14-8\*x);

}

Task7\_main.c

#include <stdio.h>

#include "func.h"

#include "task7\_func.c"

void main(){

x=5;

f();

printf("x = %.4lf\nf(x) = %.4lf", x, result);

printf("\nEnter x:\n");

scanf("%lf", &x);

f();

printf("x = %.4lf\nf(x) = %.4lf\n", x, result);

system("pause");

return 0;

}

*Task7\_main.c*

Результат аналогичен.

#### Задание 8

cmake\_minimum\_required (VERSION 3.8)

add\_executable(Task8 Task7\_main.c)

add\_library(Task8\_lib STATIC Task7\_func.c)

target\_link\_libraries (Task8 Task8\_lib)

Результат аналогичен.

# Практикум №2 Разветвляющиеся алгоритмы

### Цель практикума

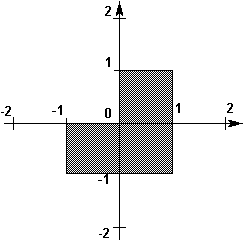
Овладеть навыками создания разветвляющихся алгоритмов на языке C с применением инструкций выбора, условных выражений, логических операций и операций отношения

### Задание

По номеру Вашего варианта выбрать задачи и выполнить следующие задания.

### Индивидуальные варианты заданий

Индивидуальный вариант для задания 1



Индивидуальный вариант для задания 2

### Решение заданий

#### Задание 1

#include <math.h>

\_Bool isInArea(double x, double y)

{

if (x\*y>0 && abs(x)\*abs(y)<=1)

return 1;

else

return 0;

}

#### Задание 2

#include <math.h>

double f(double x) //Прототип ф-и

{

if (x>3)

return (-3\*x+9);

else if (x<=3)

return (float)(pow(x,3)/(pow(x,2)+8)) ;

}

#### Задание 3

#include <stdio.h>

#include "task1.c"

#include "task2.c"

#include <locale.h>

//Вариант 10 (50 по списку)

void main(){

setlocale(LC\_ALL, "Rus");

printf("Введите номер задания:\n1 - Задание 1\n2 - Задание 2\n");

float x, y;

switch (\_getch())

{

case '1':

printf("Задание 1\nEnter x:\n");

scanf("%f", &x);

printf("Enter y:\n");

scanf("%f", &y);

isInArea(x, y) ? printf("Входит в область.\n") : printf("Не входит в область.\n");

break;

case '2':

printf("Задание 2\nEnter x:\n");

scanf("%f", &x);

printf("%f", f(x));

break;

default:

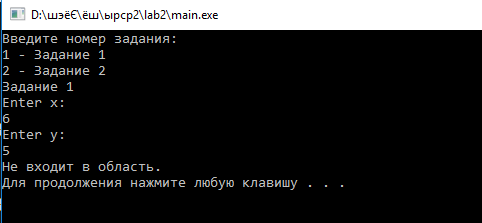
printf("Неправильно введенные данные.");

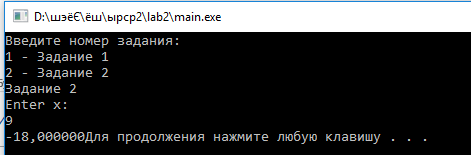
break;

}

system("pause");

}





# Практикум №3 Регулярные и итерационные циклы

### Цель практикума

Овладеть навыками создания циклических алгоритмов на языке C с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел

### Задание

По номеру Вашего варианта выбрать задачи и выполнить следующие задания.

Все указанные ниже функции разработать в трёх вариантах: с использованием цикла for, while, do … while. Функциям из задания дать одинаковые имена. Каждую функцию поместить в отдельный файл. Прототип функции вынести в заголовочный файл. В решении создать три проекта.

### Индивидуальный вариант задания

### Решение заданий (for)

#### Задание 1

#include <math.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

double summ(int n) {

int i;

double s = 0;

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

{

double a = (pow(-1,i))\*((double)(i+1)/(double)(pow(i,3)-pow(i,2)+1));

s+=a;

}

return s;

}

#### Задание 2

#include <math.h>

double summ2(double eps)

{

double s = 0;

int i;

for (i = 0; ; ++i)

{

double a = (pow(-1,i))\*((double)(i+1)/(double)(pow(i,3)-pow(i,2)+1));

if (fabs(a) <= eps)

{

return s;

}

s += a;

}

}

#### Задание 3

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

int l=0;

int i;

for (i = 0; ; ++i)

{

double a = (pow(-1,i))\*((double)(i+1)/(double)(pow(i,3)-pow(i,2)+1));

l+=1;

if (l == k)

{

l = 0;

continue;

}

printf("%.4f\n", a);

n = n - 1;

if (n == 0)

break;

}

}

#### Задание 4

#include <math.h>

int findFirstElement(double eps)

{

int i;

for (i = 0;;++i)

{

double a = (pow(-1,i))\*((double)(i+1)/(double)(pow(i,3)-pow(i,2)+1));

if (fabs(a) <= eps)

{

return i+1;

break;

}

}

}

#### Задание 5

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i;

for (i = 0;;++i)

{

double a = (pow(-1,i))\*((double)(i+1)/(double)(pow(i,3)-pow(i,2)+1));

if ((a < 0) && (fabs(a) <= eps))

{

return i+1;

}

}

}

#### Задание 6

#include <stdio.h>

#include "task1.c"

#include "task2.c"

#include "task3.c"

#include "task4.c"

#include "task5.c"

//Âàðèàíò 10 (40 ïî ñïèñêó)

int main(void)

{

int n;

int k;

int ir;

double eps;

double r;

for (;;)

{

system("cls");

printf("1 - Task1\n2 - Task2\n3 - Task3\n4 - Task4\n5 - Task5\n6 - Exit\r\n");

char answer = \_getch();

system("cls");

switch (answer)

{

case '1':

printf("\nEnter n:\n");

scanf("%i", &n);

printf("\n");

r = summ(n);

printf("%lf\n", r);

break;

case '2':

printf("\nEnter eps:\n");

scanf("%lf", &eps);

printf("\n");

r = summ2(eps);

printf("%lf\n", r);

break;

case '3':

printf("\nEnter n:\n");

scanf("%i", &n);

printf("\nEnter k:\n");

scanf("%i", &k);

printf("\n");

print(n, k);

break;

case '4':

printf("\nEnter eps:\n");

scanf("%lf", &eps);

printf("\n");

ir = findFirstElement(eps);

printf("%i\n", ir);

break;

case '5':

printf("\nEnter eps:\n");

scanf("%lf", &eps);

printf("\n");

ir = findFirstNegativeElement(eps);

printf("%i\n", ir);

break;

case '6':

return 0;

break;

default:

printf("Invalid input\n");

break;

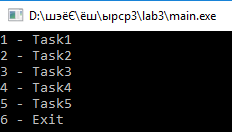
}

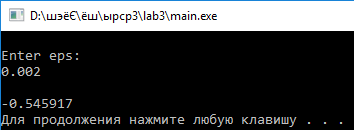
system("pause");

}

return 0;

}





Результаты с While и Do…While аналогичны

## Практикум №4 Указатели и массивы

### Цель практикума

Овладеть навыками работы с указателями и массивами в языке программирвоания C

### Индивидуальный вариант задания

Пузырьковая сортировка

### Решение заданий

#### Задание 1

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

void f (const double \*x, double \*result)

{

\*result = (double) 1/4-1/4\*sin(5/2\*3.14\*\*x);

}

void main ()

{

double x = 5;

double result;

f(&x,&result);

printf("x = %.4lf\n(x) = %.4lf", x, result);

printf("\nEnter x:\n");

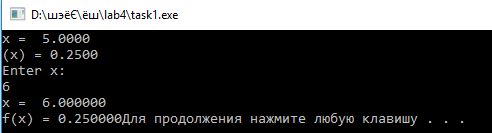
scanf("%lf", &x);

f(&x,&result);

printf("x = %f\nf(x) = %f",x, result);

system ("pause");

}



#### Задание 2

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

void writeArray (int \*arr, int n, int r) {

int i;

for (i=0;i<r;++i)

{

printf("%i ",arr[i]);

}

system("pause");

}

void main()

{

int i;

int n;

int r;

printf("first n =");

scanf("%i",&n);

printf("array range =");

scanf("%i",&r);

int a[r];

for (i=0;i<r;++i)

{

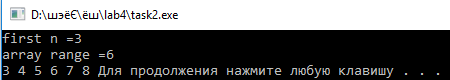
a[i] =n;

++n;

}

writeArray(a,n,r);

}



#### Задание 3

#include <stdio.h>

#include <math.h>

void readArray(int \*arr, int n)

{

int i;

printf("Input %d elements:\n", n);

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

{

scanf("%d", &arr[i]);

}

}

void main()

{

const int n = 6;

int a[n];

readArray(a, n);

system("pause");

return 0;

}

#### 

#### Задание 4

#### #include <stdio.h>

#### #include <math.h>

#### void ShellSort(int a[], int n)

#### {

#### int i, j, step;

#### int tmp;

#### for (step = n / 2; step > 0; step /= 2)

#### {

#### for (i = step; i < n; i++)

#### {

#### tmp = a[i];

#### for (j = i; j >= step; j -= step)

#### {

#### if (tmp < a[j - step])

#### a[j] = a[j - step];

#### else

#### break;

#### }

#### a[j] = tmp;

#### }

#### void main()

#### {

#### const int n = 6;

#### int a[]={4,3,6,2,1,5};

#### int i;

#### ShellSort(a, n);

#### system("pause");

#### return 0;

#### }

#### Задание 5

void sortStackArray(void) {

printf("%d\n", N);

int arr[N];

readArray(&arr, N);

writeArray(&arr, N);

sort(&arr, N);

writeArray(&arr, N);

}

#### Задание 6

#include <stdio.h>

#include <windows.system.h>

void main() {

while (1) {

sortStackArray();

printf("One more time?\n1-yes, 0-no");

int response = 0;

int\* respp = &response;

scanf("%d", respp);

if (\*respp == 0) {

exit(0);

}

}

}

## Практикум №5 Строки

### Цель практикума

Овладеть навыками работы со строками в языке программирвоания C

### Задание

#### Задание 1

int strlen(const char\* str) {

int count = 0;

for (char\* cur = str; \*cur!="\0"; ++cur) {

++count;

}

return count;

}

#### Задание 2

char\* find(const char\* str, const char\* substr) {

int lenStr = strlen(str);

int lenSub = strlen(substr);

for (int i = 0; i <= lenStr - lenSub; ++i) {

int j;

for (j = 0; str[i + j] == substr[j]; ++j);

if (j - lenSub == 1 && i == lenStr - lenSub) return &str[i];

}

return 0;

}

#### Задание 3

void delete(char\* str, char\* substr) {

char\* firstSymb = find(str, substr);

char\* lastSymb = find(str, substr) + strlen(substr);

for (char\* curr = find(str, substr); curr!=lastSymb;++curr) {

\*curr = "0";

}

}

#### Задание 4

void strcpy(char\* dest, int len, const char\* src) {

if (strlen(src) > len) {

int i;

for (i = 0; i < len - 1; ++i) {

dest[i] = src[i];

}

dest[i + 1] = "\0";

}

else

{

while ((\*dest++ = \*src++) != '\0');

}

}

#### Задание 5

void strcat(char\* dest, int len, const char\* first, const char\* second) {

int i;

dest[len - 1] = "\0";

for (i = 0; dest[i] != "\0"; ++i) {

if (first[i] != "\0") {

dest[i] = first[i];

}

else

{

dest[i] = second[i];

}

}

}

#### Задание 6

\_Bool insert(const char\* src, const char\* str, int index, char\* dest, int len) {

if (index < 0 || index > strlen(src) - 1) {

return 0;

}

else {

int i;

dest[len - 1] = "\0";

for (i = 0; dest[i] != "\0"; ++i) {

if (dest[i] = index) {

int j,k;

for (j = i,k=0; dest[j] != "\0" || j == index + strlen(str) - 2; ++j,++k, i = j) {

dest[j] = str[k];

}

}

dest[i] = src[i];

}

}

}

## Практикум №6 Структуры, объединения, перечисления и битовые поля

### Цель практикума

Овладеть навыками работы со структурами, объединениями, перечислениями и битовыми полями в языке C.

### Задание

#### Задание 1

struct Zachetka {

char firstName[30];

char lastName[30];

char middleName[40];

int number;

struct Term termsInfo[12];

};

struct Term

{

int number;

int year;

struct Subject subjectInfo[20];

};

struct Subject

{

char name[25];

\_Bool markType;///1 for exam 0 for credit

\_Bool creditMark;

enum Mark examMark;

char date[20];

char teacherLastName[30];

}

## Практикум №7 Динамическое выделение памяти

### Цель практикума

Овладеть навыками выделения динамической памяти в языке программирования C.

### Задание

#### Задание 1

void incrementHeapVariable() {

int\* p = (int\*)malloc(sizeof(int));

\*p = 10;

printf("%p %d", p, \*p);

\*p += 1;

printf("%p %d", p, \*p);

}

#### Задание 2

void sortHeapArray(int\* arr, int n) {

int\* arrayElements;

printf("Enter number of elements in array:\n");

scanf("%d", arrayElements);

\*arr = (int\*)malloc(\*arrayElements \* sizeof(int));

readArray(arr, \*arrayElements);

writeArray(arr, \*arrayElements);

sortHeapArray(arr, \*arrayElements);

writeArray(arr, \*arrayElements);

free(arr);

}

#### Задание 3

char\* heapDelete(char\* str, const char\* substr) {

char\* inHeapStr = heapStrcpy(str);

char\* inHeapSubstr = heapStrcpy(substr);

delete(inHeapStr, inHeapSubstr);

return inHeapStr;

}

#### Задание 4

char\* heapStrcpy(const char\* src) {

char\* heapStr = (char\*)(malloc(sizeof(char) \* strlen(src)));

strcpy(heapStr, strlen(src), src);

return heapStr;

}

#### Задание 5

char\* strcat(const char\* first, const char\* second) {

int len1 = strlen(s1);

int len2 = strlen(s2);

char\* result = malloc(len1 + len2 + 1);

memcpy(result, s1, len1);

memcpy(result + len1, s2, len2 + 1);

return result;

}

#### Задание 6

Написать функцию вставки подстроки в строку. Функция принимает исходнную строку, строку для вставки, индекс, определяющий место вставки, выделяет необходимое количество памяти в области динамической памяти, копирует реузльтат и возвращает указатель на вновь выделеную память.

Прототип функции char \*insert(const char \*src, const char \*str, int index).

Функция не должна использовать функции консольного ввода-вывода.

#### Задание 7

Модифицировать приложение из практикума №6, заменив статические массивы динамическими. Размеры всех массивов определяются количеством записей (количество семестров равно количеству сессий, включая текущую, количество предметов равно количеству сданных на текущий момент экзаменов и зачётов, все строковые массивы должны иметь длинну, достаточную для хранения всех символов строки, включая нулевой символ).