**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:

Крюков А.В.

Москва 2017

# 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>

void main() {

double x = 1.2;

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

printf("f(x)=%lf\n", (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x)));

printf("Input x:\n");

double x2;

scanf("%lf", &x2);

printf("x=%lf\n", x2);

printf("f(x)=%lf\n", (1 - 2 \* pow(sin(x2), 2)) / (1 + sin(2 \* x2)));

}



#### Задание 2

#include <stdio.h>

#include <math.h>

double f(double x) {

double res;

res = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

return res;

}

void main() {

double x = 1.2;

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

printf("f(x)=%lf\n", f(x));

printf("Input x:\n");

scanf("%lf", &x);

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

printf("f(x)=%lf\n", f(x));

}

Результат аналогичен.

#### Задание 3

#include <stdio.h>

#include <math.h>

void main() {

double x = 1.2;

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

printf("f(x)=%lf\n", func(x));

printf("Input x:\n");

scanf("%lf", &x);

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

printf("f(x)=%lf\n", func(x));

}

double f(double x) {

double res;

res = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

return res;

}

Результат аналогичен.

#### Задание 4

#include <stdio.h>

#include <math.h>

double x, result;

void main() {

x = 1.2;

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

printf("f(x)=%lf\n", f());

printf("Input x:\n");

scanf("%lf", &x);

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

printf("f(x)=%lf\n", f());

}

void f(void) {

result = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

}

Результат аналогичен.

#### Задание 5

#include <math.h>

double f(double x) {

double res;

res = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

return res;

}

*Task5\_func.c*

#include <stdio.h>

void main() {

double x = 1.2;

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

printf("f(x)=%lf\n", f(x));

printf("Input x:\n");

scanf("%lf", &x);

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

printf("f(x)=%lf\n", f(x));

}

*Task5\_main..c*

Результат аналогичен.

#### Задание 6

#include <math.h>

double x, result;

void f(void) {

result = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

}

*Task6\_func..c*

#include <stdio.h>

void main() {

x = 1.2;

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

f();

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

printf("Input x:\n");

scanf("%lf", &x);

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

f();

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

}

*Task6\_main..c*

Результат аналогичен.

#### Задание 7

double x, result;

void f(void);

*func.h*

#include <math.h>

#include "func.h"

void f(void) {

result = (1 - 2 \* pow(sin(x), 2)) / (1 + sin(2 \* x));

}

*Task7\_func.c*

#include <stdio.h>

#include "func.h"

void main() {

x = 1.2;

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

f();

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

printf("Input x:\n");

scanf("%lf", &x);

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

f();

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

}

*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)

*CMakeLists.txt*

Результат аналогичен.

# Практикум №2 Разветвляющиеся алгоритмы

### Цель практикума

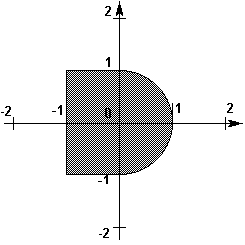
Овладеть навыками создания разветвляющихся алгоритмов на языке C с применением инструкций выбора, условных выражений, логических операций и операций отношения

### 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<=0 && x>=-1) && (y>=0 && y<=1))

return 1;

else if ((x<=0 && x>=-1) && (y<=0 && y>=-1))

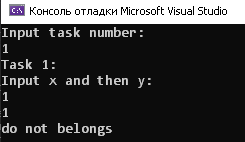
return 1;

else if ((x >= 0 && x <= 1) && (pow(x, 2) + pow(y, 2) <= 1))

return 1;

return 0;

}



#### Задание 2

#include <math.h>

double f(double x) {

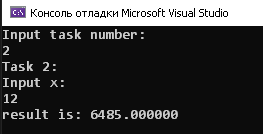
if (x >= -3.5)

return 45 \* pow(x, 2) + 5;

else

return -(2 \* x) / (-4 \* x + 1);

}



#### Задание 3

#include <stdio.h>

#include "isInArea.c"

void main() {

printf("Input task number:\n");

switch (getchar())

{

case '1':

printf("Task 1:\n");

double x, y;

printf("Input x and then y:\n");

scanf("%lf", &x);

scanf("%lf", &y);

if (isInArea(x, y)) {

printf("belongs\n");

}

else {

printf("do not belongs\n");

}

break;

case '2':

printf("Task 2:\nInput x:\n");

scanf("%lf", &x);

printf("result is: %lf\n", f(x));

break;

default:

printf("wrong task number\n");

}

}

# 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

#### Задание 1

#include "functions.h"

double summ(int n) {

double sum = 0;

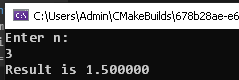
for (int i = 0; i < n; ++i) {

sum += function(i);

}

return sum;

}



#### Задание 2

#include <math.h>

#include "functions.h"

double summ2(double eps) {

double sum = 0;

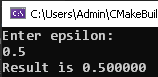
for (int i = 0; abs(function(i)) > eps; ++i) {

sum += function(i);

}

return sum;

}



#### Задание 3

#include <stdio.h>

#include "functions.h"

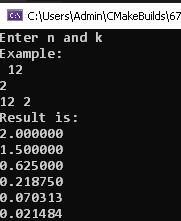
void print(int n, int k) {

for (int i = 0; i < n; i=i+k) {

printf("%lf\n", function(i));

}

}



#### Задание 4

#include <math.h>

#include "functions.h"

int findFirstElement(double eps) {

int i;

for (i = 0; 1; ++i) {

if (abs(function(i)) <= eps) {

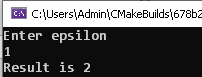
break;

}

}

return i;

}



#### Задание 5

#include <math.h>

#include "functions.h"

int findFirstNegativeElement(double eps) {

int i;

for (i = 0;1; i++) {

if (abs(function(i)) <= eps && function(i) < 0) {

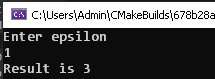
break;

}

}

return i;

}



#### Задание 6

#include <stdio.h>

#include <windows.system.h>

#include "functions.h"

void main() {

for (int i = 0; i >= 0; i++) {

//menu output

for (int i = 1;i < 7;i++) {

if (i == 6) {

printf("%d. Exit\n", i);

}

else {

printf("%d. Task %d\n", i, i);

}

}

printf("Enter task number:\n");

int response=1;

int\* responsep = &response;

scanf("%d", responsep);

switch (response)

{

case 1:

system("cls");

printf("Enter n:\n");

int n;

int\* np = &n;

scanf("%d", np);

printf("Result is %lf\n", summ(\*np));

break;

case 2:

system("cls");

printf("Enter epsilon:\n");

double eps;

double\* epsp = &eps;

scanf("%lf", epsp);

printf("Result is %lf\n", summ2(\*epsp));

break;

case 3:

system("cls");

printf("Enter n and k\nExample:\n 12 \n2\n");

int numberElements;

int\* numberElementsp = &numberElements;

int k;

int\* kp = &k;

scanf("%d", numberElementsp);

scanf("%d", kp);

printf("Result is:\n");

print(\*numberElementsp, \*kp);

break;

case 4:

system("cls");

printf("Enter epsilon\n");

double epsilon;

double\* epsilonp = &epsilon;

scanf("%lf", epsilonp);

printf("Result is %d\n", findFirstElement(\*epsilonp));

break;

case 5:

system("cls");

printf("Enter epsilon\n");

double epsilon2;

double\* epsilonp2 = &epsilon2;

scanf("%lf", epsilonp2);

printf("Resultis %d\n", findFirstNegativeElement(\*epsilonp2));

break;

default:

break;

}

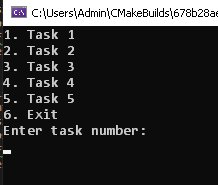
if (\*responsep == 6) {

exit(0);

}

}

}



## 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 <math.h>

void f(const double\* px, double\* resultp) {

\*resultp = (1 - 2 \* pow(sin(\*px), 2)) / (1 + sin(2 \* \*px));

}

void main() {

const double x = 1.2;

const double\* px = &x;

double result = 0;

double\* resultp = &result;

printf("x=%lf\n", \*px);

f(px, resultp);

printf("f(x)=%lf\n", \*resultp);

printf("Input x:\n");

const double x2;

double\* px2 = &x2;

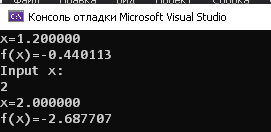
scanf("%lf", px2);

printf("x=%lf\n", \*px2);

f(px2, resultp);

printf("f(x)=%lf\n", \*resultp);

}



#### Задание 2

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

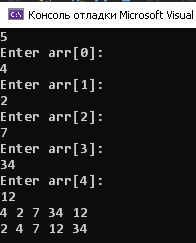
for (int\* cur = arr; cur - arr < n; ++cur) {

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

}

printf("\n");

}



#### Задание 3

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

for (int\* cur = arr; cur - arr < n; ++cur) {

printf("Enter arr[%d]:\n", cur-arr);

scanf("%d", cur);

}

}

#### Задание 4

//bubble sort

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

for (int i = n - 1; i >= 0; i--)

{

for (int j = 0; j < i; j++)

{

if (\*(arr+j) > \*(arr+j+1))

{

int tmp = \*(arr + j);

\*(arr + j) = \*(arr + j + 1);

\*(arr + j + 1) = tmp;

}

}

}

}

#### 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];

}

}

}

#### Задание 7

## 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(first);

int len2 = strlen(second);

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

memcpy(result, first, len1);

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

return result;

}

#### Задание 6

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

char\* strC;

strC = (char\*)malloc(strlen(destination) + strlen(seed) + 1);

strncpy(strC, src, index);

strC[index] = '\0';

strcat(strC, index);

strcat(strC, src + index);

strcpy(src, strC);

free(strC);

}

#### Задание 7

#include <stdlib.h>

struct Zachetka {

char firstName = (char\*)malloc(sizeof(char)\*30);

char lastName = (char\*)malloc(sizeof(char) \* 30);

char middleName = (char\*)malloc(sizeof(char) \* 40);

int number;

struct Term\* termsInfo = malloc(sizeof(\*termsInfo)\*4);

};

struct Term

{

int number;

int year;

struct Subject\* subjectInfo = malloc(sizeof(\*subjectInfo)\*2);

};

struct Subject

{

char name = (char\*)malloc(sizeof(char) \* 30);

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

\_Bool creditMark;

enum Mark examMark;

char date = (char\*)malloc(sizeof(char) \* 30);

char teacherLastName = (char\*)malloc(sizeof(char) \* 30);

};