**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра «Информационная безопасность»

**ЛАБОРАТОРНАЯ РАБОТА**

**№3**

по дисциплине «Объектно-ориентированное программирование систем защиты информации»

на тему:

**«Регулярные и итерационные циклы»**

Вариант №10

Выполнил: студ. гр. БПЗ1501

Лисеенко Я.А.

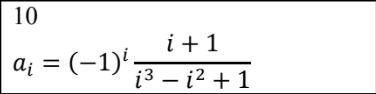
Проверил: ассистент кафедры ИБ

Барков В.В.

Москва 2018 г

**Цель:** овладеть навыками создания циклических алгоритмов на языке C с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.

**Задание:**



**Выполнение:**

**Задание 1:**

**For:**

#include <math.h>

double summ(int n) {

double a = 1;

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

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

}

return a;

}

**While:**

#include <math.h>

double summ(int n) {

double a = 1;

int i = 1;

while (i <= n) {

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

i++;

}

return a;

}

**Do … while:**

#include <math.h>

double summ(int n) {

double a = 1;

int i = 1;

do {

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

i++;

} while (i <= n);

return a;

}

**Задание 2:**

**For:**

#include <math.h>

double summ2(float eps) {

float a0 = 1, a1;

int i = 1;

for (;;) {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

if (fabs(a1 - a0) <= eps)

break;

a0 = a1;

i++;

}

return a1;

}

**While:**

#include <math.h>

double summ2(float eps) {

float a0 = 1, a1 = 0, diff = fabs(a0 - a1);

int i = 1;

while (diff > eps) {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

diff = fabs(a0 - a1);

a0 = a1;

i++;

}

return a1;

}

**Do … while:**

#include <math.h>

double summ2(float eps) {

float a0 = 1, a1 = 0, diff = fabs(a0 - a1);

int i = 1;

do {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

diff = fabs(a0 - a1);

a0 = a1;

i++;

} while (diff > eps);

return a1;

}

**Задание 3:**

**For:**

#include <math.h>

void print(int n, int k) {

float a = 1;

printf("%i ", 0);

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

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

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

if (i%k == 0)

continue;

printf("%i " ,i);

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

}

}

**While:**

#include <math.h>

void print(int n, int k) {

float a = 1;

printf("%i ", 0);

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

int i = 1;

while (i < n) {

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

if (i % k == 0) {

i++;

continue;

}

printf("%i ", i);

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

i++;

}

}

**Do … while:**

#include <math.h>

void print(int n, int k) {

float a = 1;

printf("%i ", 0);

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

int i = 1;

do {

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

if (i % k == 0) {

i++;

continue;

}

printf("%i ", i);

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

i++;

} while (i < 20);

}

**Задание 4:**

**For:**

#include <math.h>;

int findFirstElement(float eps) {

float a0 = 1, a1 = 0;

int i = 1;

for (;;) {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

if (fabs(a1 - a0) < eps) {

return i;

break;

}

i++;

}

}

**While:**

#include <math.h>;

int findFirstElement(float eps) {

float a0 = 1, a1 = 0, diff = fabs(a0 - a1);

int i = 1;

while (diff > eps) {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

diff = fabs(a0 - a1);

a0 = a1;

++i;

}

i--;

return i;

}

**Do … while:**

#include <math.h>;

int findFirstElement(float eps) {

float a0 = 1, a1 = 0, diff = fabs(a0 - a1);

int i = 1;

do {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

diff = fabs(a0 - a1);

a0 = a1;

++i;

} while (diff > eps);

i--;

return i;

}

**Задание 5:**

**For:**

#include <math.h>;

int findFirstElement(float eps) {

float a0 = 1, a1 = 0, diff = fabs(a0 - a1);

int i = 1;

do {

a1 = a0 + pow((-1), i)\*(pow(2, i + 1) / (pow(2, 2 \* i) + 1));

diff = fabs(a0 - a1);

a0 = a1;

++i;

} while (diff > eps);

i--;

return i;

}

**While:**

#include <math.h>;

int findFirstNegativeElement(float eps) {

float a0 = 1, a1 = 0;

int i = 1;

while (i != 0) {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

if (fabs(a1 - a0) < eps && a1 < 0) {

return i;

break;

}

else {

i = 0;

return i;

break;

}

a0 = a1;

i++;

}

}

**Do … while:**

#include <math.h>;

int findFirstNegativeElement(float eps) {

float a0 = 1, a1 = 0;

int i = 1;

do {

a1 = a0 + pow(-1, i)\*((i + 1) / (pow(i, 3) - pow(i, 2) + 1));

if (fabs(a1 - a0) < eps && a1 < 0) {

return i;

break;

}

else {

i = 0;

return i;

break;

}

a0 = a1;

i++;

} while (i != 0);

}

**Задание 6:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "Header.h"

int main() {

float eps;

int n, k;

double a;

for (int i = 0; i != -1;) {

system("cls");

printf("task 1 \n");

printf("task 2 \n");

printf("task 3 \n");

printf("task 4 \n");

printf("task 5 \n");

printf("6 to Exit \n \n");

printf("Choose Your Destiny");

switch (\_getch()) {

case '1':

system("cls");

printf("task 1 \n");

printf("n = ");

scanf("%i", &n);

a = summ(n);

printf("%f", a);

\_getch();

break;

case '2':

system("cls");

printf("task 2 \n");

printf("eps = ");

scanf("%f", &eps);

a = summ2(eps);

printf("%f", a);

\_getch();

break;

case '3':

system("cls");

printf("task 3 \n");

printf("n = ");

scanf("%i", &n);

printf("k = ");

scanf("%i", &k);

print(n, k);

\_getch();

break;

case '4':

system("cls");

printf("task 4 \n");

printf("eps = ");

scanf("%f", &eps);

n = findFirstElement(eps);

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

\_getch();

break;

case '5':

system("cls");

printf("task 5 \n");

printf("eps = ");

scanf("%f", &eps);

n = findFirstNegativeElement(eps);

if (n != 0) {

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

}

else {

printf("No Negative Elements");

}

\_getch();

break;

case '6':

i = -1;

break;

default:

break;

}

}

\_getch();

return 0;

}

**Результат:**

