Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт: Институт цифрового развития

Кафедра: Межинститутская базовая кафедра

**Отчёт по Лабораторной работе № 3 по информационным технологиям и программированию**

выполнил:

Студент, 1 курс, группа ПИЖ-б-о-23-2

направление подготовки

09.03.04 Программная инженерия,

направленность «Разработка и сопровождение программного обеспечения »

форма обучения очная

Черников Дмитрий Дмитриевич

Ставрополь, 2023г

Лабораторная работа № 3. Разработка программ циклической

структуры

**Цель работы**: научиться разрабатывать программы алгоритмов

циклической структуры

**Задача 1**. Задача про сумму чисел от 1 до 100

Вывести на экран все числа от 1 до 100 и их сумму.

Код программы:

#include <iostream>

using namespace std;

int main() {

int i, s;

i = 1;

s = 0;

while (i <= 100) {

cout << i << " ";

s = s + i;

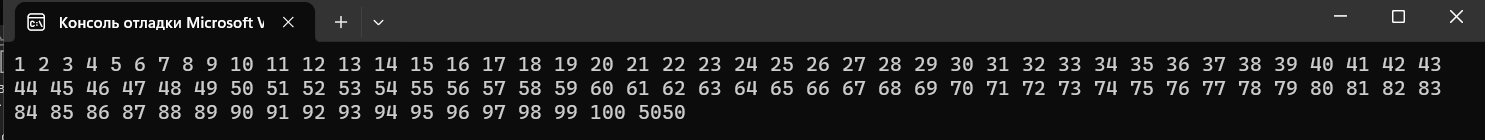
i = i + 1;

}

cout << s << endl;

return 0;

}



**Задача 2**. Задача про степень двойки

Найдём наибольшую степень двойки, которая не превосходит 1000

Код программы:

#include <iostream>

using namespace std;

int main()

{

int ans = 1;

while (ans \* 2 < 1000) {

ans = ans \* 2;

}

cout << ans;

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Задача 3**. Задача про наибольшее число в последовательности

Дана последовательность чисел, необходимо найти самое большое

число в последовательности. Признаком завершения последовательности

является число 0.

Код программы:

#include <iostream>

using namespace std;

int main()

{

int x, ma = 0;

cin >> x;

while (x != 0) {

if (x > ma) {

ma = x;

}

cin >> x;

}

cout << ma << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

**Задача 4.** Задача про цифры числа

Дано число x. Необходимо посчитать количество и сумму его цифр

Код программы:

#include <iostream>

using namespace std;

int main()

{

int x, cnt = 0, s = 0;

cin >> x;

while (x > 0) {

cnt = cnt + 1;

s = s + x % 10;

x = x / 10;

}

cout << cnt << " " << s << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задача 5**. Задача про наибольшее число в последовательности

Дана последовательнось чисел. Необходимо найти самое большое число

в последовательности. Признаком завершения последовательности является

число 0.

Код программы:

#include <iostream>

using namespace std;

int main()

{

int maxx, now;

cin >> now;

maxx = now;

while (now > 0) {

if (now > maxx) {

maxx = now;

}

cin >> now;

}

cout << maxx << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Задача 6**. Задача про количество максимумов в последовательности

Дана последовательнось чисел. Необходимо найти максимум

в последовательности и количество элементов, значение которых равно

максимуму. Признаком завершения последовательности является число 0

Код программы:

#include <iostream>

using namespace std;

int main()

{

int maxx, now, cntmax = 0;

cin >> now;

maxx = now;

while (now > 0) {

if (now > maxx) {

maxx = now;

cntmax = 1;

}

else if (now == maxx) {

cntmax = cntmax + 1;

}

cin >> now;

}

cout << maxx << " " << cntmax << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Задача 7**. Бесконечный цикл

Если в реализации программы допустить ошибку, то цикл может

работать бесконечно долго. Рассмотрим это на примере задачи, в которой

требуется вывести все числа от 1 до n

Код программы:

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

int i = 1;

while (i <= n) {

cout << i << " ";

i = i + 1;

}

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Задача 8**. Задача о локальных максимумах

Элемент последовательности называется локальным максимумом, если

он строго больше предыдущего и последующего элементов

последовательности. Первый и последний элементы последовательности не

являются локальными максимумами.

Код программы:

#include <iostream>

using namespace std;

int main()

{

int prev, now, next, cnt = 0;

cin >> prev;

cin >> now;

cin >> next;

while (next > 0) {

if (now > prev && now > next) {

cnt = cnt + 1;

}

prev = now;

now = next;

cin >> next;

}

cout << cnt << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

**Задача 9**. Задача о поиске числа

Дана последовательность и число x. Необходимо вывести номер

позиции, в которой число x первый раз встречается в последовательности.

Если число x не присутствует в последовательности, то необходимо вывести

количество элементов в ней. Признаком завершения последовательности

является число 0.

Код программы:

#include <iostream>

using namespace std;

int main()

{

int x, now, i = 0;

cin >> x;

cin >> now;

while (now > 0) {

if (now == x) {

break;

}

i = i + 1;

cin >> now;

}

cout << i;

return 0;

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

**Задача 10**. Задача о чётных числах

Дана последовательность. Необходимо вывести сумму всех чётных

элементов последовательности. Признаком завершения последовательности

является число 0

Код программы:

#include <iostream>

using namespace std;

int main()

{

int now, summ = 0;

cin >> now;

while (now > 0) {

if (now % 2 != 0) {

cin >> now;

continue;

}

summ = summ + now;

cin >> now;

}

cout << summ << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Задача 11**. Задача про числа от 1 до 100

Выведите на экран все числа от 1 до 100

Код программы:

#include <iostream>

using namespace std;

int main()

{

int i = 1;

while(i <= 100) {

cout << i << " ";

i++;

}

cout << endl;

for (i = 1; i <= 100; ++i)

cout << i << " ";

}

Решение через 2 вида циклов

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**12. Задача** про таблицу умножения

Выведите на экран таблицу умножения для чисел от 1 до 10

Код программы:

#include <iostream>

using namespace std;

int main()

{

int i = 0;

int j = 0;

for (i = 1; i <= 10; ++i) {

for (int j = 1; j <= 10; ++j) {

cout << i \* j << " ";

}

if (i != 10) {

cout << endl;

}

}

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**13. Задача**

Дана последовательность из n натуральных чисел. Найдите сумму

чётных чисел в последовательности. Значения всех чисел не превышают 231-1

Код программы:

#include <iostream>

using namespace std;

int main() {

long long sum = 0;

int now, n;

cin >> n;

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

cin >> now;

if (now % 2 == 0) {

sum += now;

}

}

cout << sum << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание

**14. Задача**

Даны два натуральных числа - A и B. Необходимо вывести все

нечётные числа на отрезке [A;B]. Гарантируется, что A - нечётное и A<B

Код программы:

#include <iostream>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

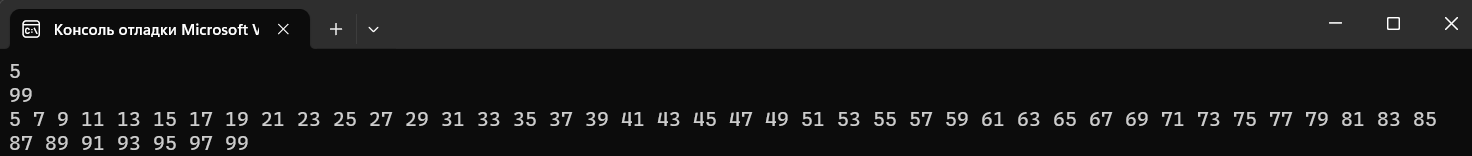
for (int i = a; i <= b; i += 2) {

cout << i << " ";

}

return 0;

}



**15. Задача**

Выведите все n-значные натуральные числа.

Код программы:

#include <iostream>

using namespace std;

int main()

{

int d;

cin >> d;

int mind = 1, maxd;

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

mind \*= 10;

}

maxd = mind \* 10 - 1;

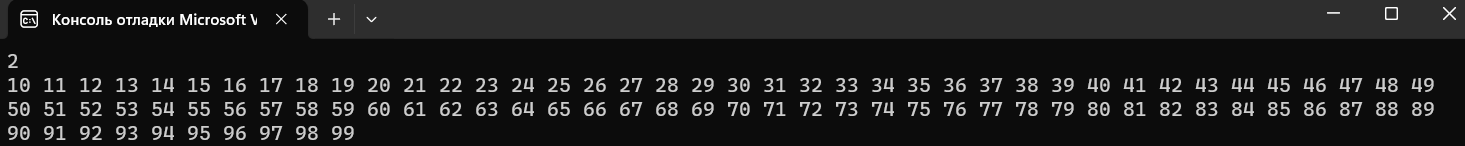
for (int i = mind; i <= maxd; ++i) {

cout << i << " ";

}

return 0;

}



**16. Задача**

Даны три натуральных числа - a, b, c. Выведите все числа на отрезке

от [a;b], делящиеся на c

Код программы:

#include <iostream>

using namespace std;

int main()

{

int a, b, c;

cin >> a >> b >> c;

int m = a % c;

if (m != 0) {

a += c - m;

}

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

cout << i << " ";

}

cout << endl;

return 0;

}

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Автоматически созданное описание

**4.** **Самостоятельно разработать программы к следующим задачам**

**по вариантам**

**Вариант – 11(1)**

**Задание 1**. Разработать программу на языке программирования С++

по вариантам

Цифры числа - Дано 10-значное число. Выведите все цифры этого числа в обратном порядке по одной.

Входные данные - На вход подаётся натуральное 10-значное число.

Выходные данные - Выведите ответ на задачу. В качестве разделителя между

цифрами можно использовать переводы строки и пробелы.

Код программы:

#include <iostream>

#include <string>

#include <math.h>

using namespace std;

int main() {

int num;

cin >> num;

string result;

while (true) {

result += to\_string(num%10) ;

num = num / 10;

if (num == 0) {

break;

}

result += " ";

}

cout << result;

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Задание 2.**

Составить программу расчета таблицы значений функции:

Y = 5(1-e-0,5t)cos(2 t), t >= 0

в интервале a t b в n равностоящих точках. Границы интервала [a,b] и количество точек n ввести с клавиатуры. Результаты вывести на экран

Код программы:

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <string>

#include <math.h>

#include <cmath>

using namespace std;

int main() {

double Y;

double e = M\_E;

double pi = M\_PI;

double t;

double a, b;

int n;

cin >> a >> b >> n;

t = a;

for (; t <= b; t += (b - a) / (n-1)) {

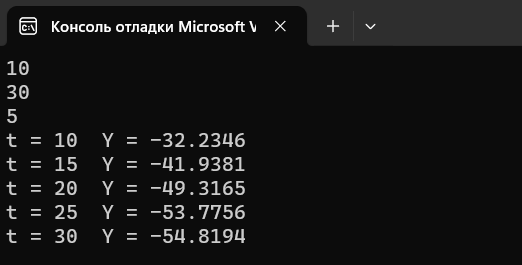
Y = 5 \* (1 - e - 0.5 \* t) \* cos(pow(2, pi) \* t);

cout << "t = " << t << " " << "Y = " << Y << endl;

}

return 0;

}



**Задание 3.**

Определите наименьшее расстояние между двумя локальными максимумами последовательности натуральных чисел, завершающейся числом 0. Если в последовательности нет двух локальных максимумов, выведите число 0. Начальное и конечное значение при этом локальными максимумами не считаются.

Расстоянием считается количество пробелов между элементами.

Формат входных данных Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).

Формат выходных данных Выведите ответ на задачу

Код программы:

#include<iostream>

using namespace std;

int main() {

int len = 0;

int maxlen = 0;

int preprev;

int prev;

int now;

cin >> preprev >> prev;

while (true) {

cin >> now;

if (now == 0) {

break;

}

len++;

if (prev > preprev && prev > now) {

len--;

if (len > maxlen) {

maxlen = len;

}

len = 0;

}

preprev = prev;

prev = now;

}

cout << maxlen;

}

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Контрольные вопросы**

1. **Какой алгоритм называется циклическим?**

Циклический алгоритм — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций). К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно.

1. **Какие типы алгоритмов существуют? Опишите принцип их работы?**

1. Рекурсивные алгоритмы — работают путем повторного вызова функции до тех пор, пока не будет достигнуто определенное условие. Принцип работы заключается в том, что функция вызывает саму себя до тех пор, пока не выполнится заданное условие.

2. Итеративные алгоритмы — работают путем последовательного выполнения шагов до достижения определенного результата. Принцип работы заключается в том, что функция выполняет один шаг и затем проверяет, было ли достигнуто заданное условие. Если нет, то функция продолжает выполнение следующего шага.

3. Последовательные алгоритмы — работают путем последовательной обработки элементов входных данных. Принцип работы заключается в том, что функция обрабатывает каждый элемент входных данных по очереди до достижения определенного результата.

4. Параллельные алгоритмы — работают путем одновременной обработки нескольких элементов входных данных. Принцип работы заключается в том, что функция разбивает входные данные на несколько частей и обрабатывает каждую часть одновременно.

5. Циклические алгоритмы также являются одним из типов алгоритмов. Они работают путем повторения одного и того же блока кода до тех пор, пока не будет достигнуто определенное условие. Принцип работы заключается в том, что функция выполняет один блок кода и затем проверяет, было ли достигнуто заданное условие. Если нет, то функция продолжает выполнение следующего блока кода.

**3. Опишите операторы цикла на С++**

В C++ существует несколько операторов цикла, которые позволяют повторять блок кода до тех пор, пока не будет достигнуто определенное условие. Некоторые из них:

1. Цикл while — используется для повторения блока кода до тех пор, пока не будет достигнуто определенное условие.
2. Цикл do-while — используется для повторения блока кода хотя бы один раз, даже если условие не выполняется.
3. Цикл for — используется для повторения блока кода определенное количество раз.
4. **Чем преинкремент отличается от постинкремента?**

Основное различие между операторами постинкремента и преинкремента заключаются в следующем: При постинкременте вызывающий код в начале получает значение переменной, а затем значение переменной увеличивается на единицу. При преинкременте в начале значение переменной увеличивается на единицу, а затем вызывающий код получает значение переменной.