Бюджетное учреждение высшего образования   
 Ханты-Мансийского автономного округа   
 «Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 4 «Одномерные массивы»

по дисциплине «Программирование и основы алгоритмизации»

Выполнил: Хайитов Ш. Д.

студент группы 605-31

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2024 г.

Цель работы:

* Закрепление теоретических знаний об одномерных массивах;
* Получение практических навыков использования массивов;
* Закрепление знаний и навыков использования условных и циклических операторов;

Задание:

Заданы размер массива, действительные значения его элементов и количество элементов в группе (от 2 до 15). Дополнить каждую группу исходного массива еще одним элементом, который содержит наибольший общий делитель элементов группы. Неполная группа должна быть дополнена нулями.

**Вариант 26**

**Формальное описание задачи.**

Пользователь задает размер массива, количество элементов в одной группе. Потом пользователь вводит сами элементы.

Представим, пользователь задал размер массива 4, а количество элементов в группе 2.

Имеем:  
[1, 2, 3, 4]

Тут массив делится на две группы и у каждой группы ищется НОД и пишется в конце

Должны получить:  
[1, 2, 1, 3, 4, 1]

Представим, пользователь задал размер массива 5, а количество элементов в группе 2.

Имеем:  
[1, 2, 3, 4, 5]

Неполную группу мы дополняем нулем.

Дополнение группы:  
[1, 2, 3, 4, 5, 0]

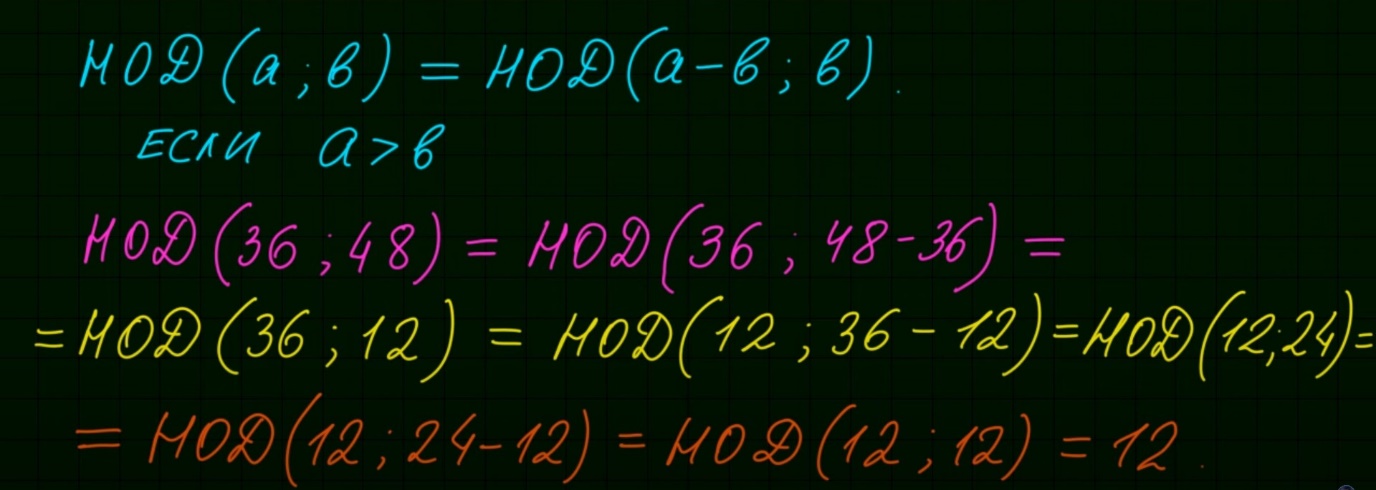
Теперь мы имеем полную группу. Находим НОД.

Итог:  
[1, 2, 1, 3, 4, 1, 5, 0, 5]

Мы будем использовать алгоритм Евклида для нахождения НОД.

Алгоритм Евклида — это алгоритм для нахождения наибольшего общего делителя (НОД) двух чисел. Он основан на принципе, что если два числа имеют общий делитель, то этот делитель также является делителем их разности. Поэтому, чтобы найти НОД двух чисел, нужно продолжать вычитать из большего числа меньшее до тех пор, пока они не станут равными или пока одно из них не станет равным нулю.

Пример:

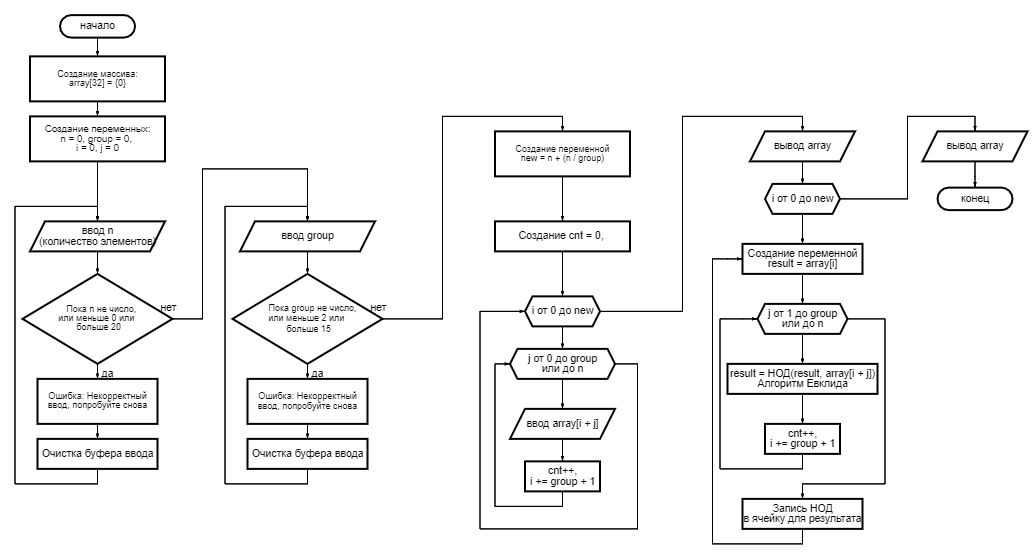


**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввод: размер массива, размера групп и значения.
2. Вычисление количества групп и количества элементов в группе.
3. Заполнение группы недостающими элементами (нулями)
4. Вычисление НОД.
5. Добавление НОД в конец группы.
6. Вывод полученного массива с добавленными элементами.

**Блок-схема**

****

**Листинг программы**

\*/

Лабораторная работа 4

Вариант 26

Заданы размер массива, действительные значения его элементов и количество элементов в группе (от 2 до 15). Дополнить каждую группу исходного массива еще одним элементом, который содержит наибольший общий делитель элементов группы. Неполная группа должна быть дополнена нулями.

\*/

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <stdlib.h>

int main() {

//Задаем размер массиву

double array[32] = {0};

int n = 0, group = 0;

setlocale(LC\_ALL, "rus");

// Получаем количество элементов в массиве

printf("Введите кол-во элементов (Не больше 20): ");

while (scanf("%d", &n) != 1 || n < 0 || n > 20) {

printf("Некорректный ввод, попробуй снова\n");

while (getchar() != '\n'); // очистка буфера ввода

}

// Получаем количество элементов в группе

printf("Введите кол-во элементов в группе (Не больше 15): ");

while (scanf("%d", &group) != 1 || group < 2 || group > 15) {

printf("Некорректный ввод, попробуй снова\n");

while (getchar() != '\n');

}

int new = n + (n / group);

// Получаем сами элементы

int cnt = 0;

for (i = 0; i < new; i += group + 1) {

for (j = 0; j < group && cnt != n; j++, cnt++) {

printf("Введите %d значение \n", cnt + 1);

if (!scanf("%lf", &array[i + j])) {

char ch;

printf("Некорекный ввод, попробуй снова\n");

while (((ch = getchar()) != '\n') && (ch != EOF));

j--, cnt--;

}

}

}

// Выводим исходный массив

printf("Исходный массив:\n");

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

printf("%.2lf ", array[j]);

}

putchar('\n');//просто printf только для одного символа

// обработка группы

cnt = 0;

for (i = 0; i < new; i += group + 1) {

double result = array[i];

for (j = 1; j < group && cnt != n; j++, cnt++) {

double b = array[i + j];

// Алгоритм Евклида (для вычисления НОД)

while (result != 0 && b != 0) {

if (result > b) {

result = fmod(result, b);

} else {

b = fmod(b, result);

}

}

result += b;

}

/\*В конце обработки группы записываем результат\*/

array[i + group] = result;

}

//Выводим итоговый массив

printf("Конечный массив:\n");

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

printf("%.2lf ", array[j]);

}

putchar('\n');

system("pause");

return 0;

}

**Пояснения к программе.**

1. В начале программы объявляются необходимые переменные и массив array размером 32. Это сделано потому, что максимальное количество элементов в массиве может быть 20, и каждая группа может содержать до 15 элементов. Таким образом, всего получается 2 группы и 2 дополнительных элемента для записи результатов.
2. Далее программа запрашивает у пользователя количество элементов и количество групп. Проверка ввода осуществляется с помощью цикла while, который повторяется до тех пор, пока пользователь не введет корректные данные.
3. После ввода данных программа заполняет массив реальными числами, которые вводит пользователь. Для каждого элемента массива используется цикл for и функция scanf для чтения числа. Также предусмотрена проверка на корректность ввода с помощью while и getchar() для очистки буфера ввода.
4. Затем программа дополняет каждую группу одним элементом, содержащим наибольший общий делитель всех элементов этой группы. Для этого используется алгоритм Евклида, реализованный внутри цикла for. В конце каждой обработки группы результат записывается в массив.
5. В конце программа выводит итоговый массив, используя двойной цикл for. Один цикл проходит по всем элементам массива, включая добавленные, а второй цикл печатает элементы массива.

**Вывод:**

Закрепил теоретические знания об одномерных массивах, получил практические навыки использования массивов, закрепил знания и навыки использования условных и циклических операторов.