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

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 6 «Функции»

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 и приобретение практических навыков объявления и реализации функций языка С;
* передачи параметров и возвращаемого значения при их вызове;

Задание:

Требуется выполнить реализацию задачи в соответствии с индивидуальным заданием, выданным преподавателем по теме «Одномерные массивы», с использованием функций для обработки массивов.

**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.

**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:

Изображение выглядит как текст, рукописный текст, грифельная доска, Шрифт

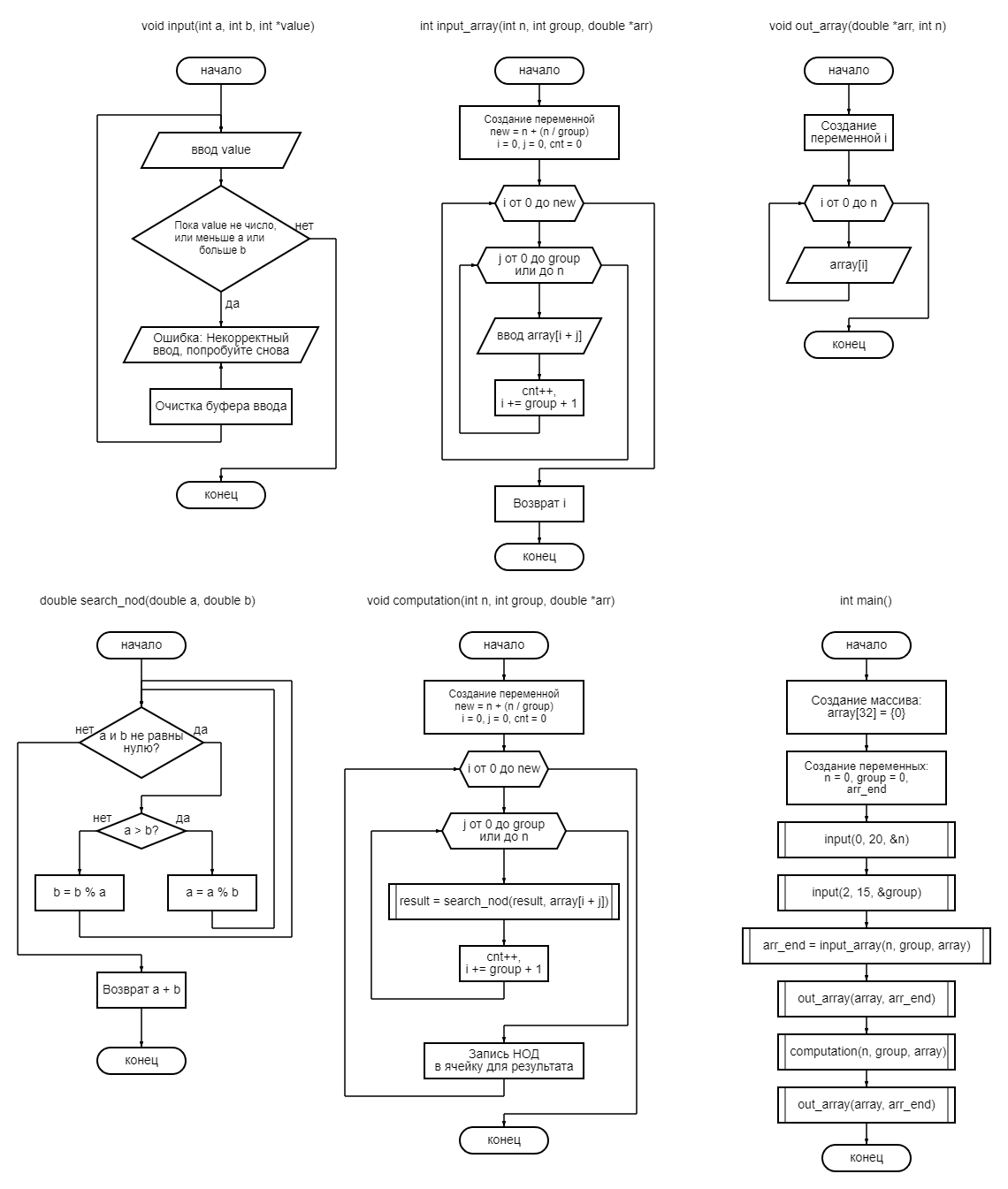
Автоматически созданное описание

**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-схемы**

****

**Листинг программы**

\*/

Лабораторная работа 6

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 <stdlib.h>

#include <locale.h>

#include <math.h>

//Записывает введённое значение в value. Устанавливает ограничение на ввод (a <=

// \* value <= b)\*/

void input(int a, int b, int \*value);

//Ввод элементов вещественного массива размера n с учётом размера групп group.

//Возвращает новый размер массива

int input\_array(int n, int group, double \*arr);

//Возвращает НОД чисел а и b. Используется алгоритм Евклида

double search\_nod(double a, double b);

//Рассчитывает НОД для каждой группы размера group в вещественном массиве arr

// \* размера n. Результат записывается в каждый элемент после группы

void computation(int n, int group, double \*arr);

//Выводит вещественный массив arr размера n в консоль

void out\_array(double \*arr, int n);

int main() {

double array[32] = {0};

int n = 0, group = 0, arr\_end = 0;

setlocale(LC\_ALL, "Rus");

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

input(0, 20, &n);

printf("Введите кол-во элементов в группе (От 2 до 15): ");

input(2, 15, &group);

arr\_end = input\_array(n, group, array);

printf("Исходный массив:\n");

out\_array(array, arr\_end);

computation(n, group, array);

printf("Конечный массив:\n");

out\_array(array, arr\_end);

return 0;

}

void input(int a, int b, int \*value) {

while (scanf("%d", &\*value) != 1 || \*value < a || \*value > b) {

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

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

}

}

int input\_array(int n, int group, double \*arr) {

int new = n + (n / group);

int i, j, 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", &arr[i + j])) {

char ch;

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

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

;

j--, cnt--;

}

}

}

return i;

}

double search\_nod(double a, double b) {

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

if (a > b) {

a = fmod(a, b);

} else {

b = fmod(b, a);

}

}

return a + b;

}

void computation(int n, int group, double \*arr) {

int new = n + (n / group);

int i, j, cnt = 0;

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

double result = arr[i];

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

result = search\_nod(result, arr[i + j]);

}

arr[i + group] = result;

}

}

void out\_array(double \*arr, int n) {

int i;

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

printf("%.2lf ", arr[i]);

}

putchar('\n');

}

**Пояснения к программе.**

1. void input(int a, int b, int \*value) - функция для безопасного ввода числовых значений. Она ограничивает ввод числом между a и b, и очищает буфер ввода, если был введен неправильный ввод.

2. int input\_array(int n, int group, double \*arr) - функция для ввода элементов вещественного массива размера n с учетом размера групп group. Возвращает новый размер массива.

3. double search\_nod(double a, double b) - функция, возвращающая наибольший общий делитель (НОД) чисел a и b. Используется алгоритм Евклида.

4. void computation(int n, int group, double \*arr) - функция, рассчитывающая НОД для каждой группы размера group в вещественном массиве arr размера n. Результат записывается в каждый элемент после группы.

5. void out\_array(double \*arr, int n) - функция для вывода вещественного массива arr размера n в консоль.

6. main() - точка входа программы. Здесь происходит инициализация переменных, ввод данных, обработка данных и вывод результатов.

**Вывод:**

Закрепил теоретические знания и приобрел практические навыки объявления и реализации функций. Решил задание из 4 лабораторной работы при помощи функций.