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: Комбаев И. Д.

студент группы 609-01

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2021 г.

Цель работы:

* закрепление теоретических знаний об одномерных массивах;
* приобретение практических навыков использования массивов;
* закрепление знаний и навыков использования условных и циклических операторов

Задание: решить задачу в соответствии с индивидуальным заданием, выданным преподавателем, с использованием в реализации алгоритма одномерных массивов.

Вариант 17. Задан размер массива вещественных чисел и значения его элементов. Определить, являются ли его элементы членами арифметической прогрессии и вывести значение разности прогрессии. Элементы исходного массива могут быть не упорядочены

**Вариант 17**

**Формальное описание задачи.**

Цель решения – определить, являются ли введенные элементы членами арифметической прогрессии.

Арифметическая прогрессия - числовая последовательность, каждый член которой, начиная со второго, равен сумме предыдущего члена и одного и того же числа d. Таким образом каждый элемент арифметической прогрессии задается формулой , где N0 – Первый элемент арифметической прогрессии, d – шаг арифметической прогрессии.

Шаг арифметической прогрессии по определению является разницей между **соседними элементами арифметической прогрессии.** , где i – номер элемента арифметической последовательности.

Зная всё это можно выделить условие для соответствия числовой последовательности определению арифметическая прогрессия:

При упорядоченной числовой последовательности: **Разница по модулю между всеми соседними элементами одинакова**

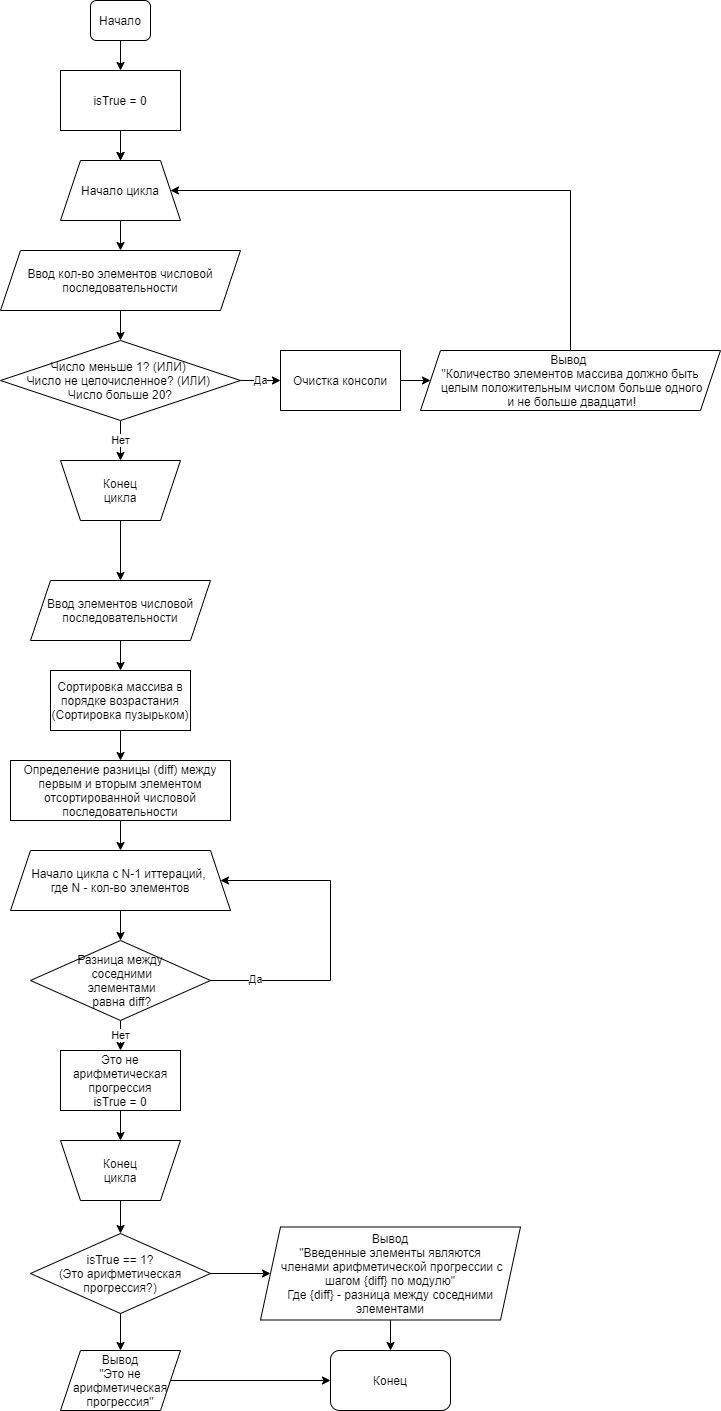
Таким образом, если числовая последовательность будет иметь разницу между одними соседними элементами, **отличающиеся** от разницы между другими соседними элементами – то эта числовая последовательность **не является арифметической прогрессией**

Поэтому решение задачи состоит в упорядочивание числовой последовательности и проверить – являются все разности между соседними элементами равными.

**Алгоритм решения:**

1. Ввести исходные данные – Кол-во элементов числовой последовательности, её элементы
2. Проверить введенные значения исходя из условий задачи
3. Упорядочить арифметическую прогрессию
4. Проверить равенство всех разниц между соседними элементами
5. Вывести результат

**Блок схема**



**Листинг программы.**

1. /\*  Задан размер массива вещественных чисел и значения его элементов.
2. Определить, являются ли его элементы членами арифметической прогрессии и вывести значение разности прогрессии.
3. Элементы исходного массива могут быть не упорядочены.\*/
5. #include <stdio.h>
6. #include <stdlib.h>
7. #include <math.h>
9. main() {
10. **int** i, isTrue = 1, j;
11. **double** numbers[20], diff, len;
12. system("chcp 1251 > nul");
13. // Если введенное число отрицательно или не целое - это неправильный ввод
14. **while**(1) {
15. printf("Введите размер массива: ");
16. scanf("%lg", &len);
17. **if** (len <= 1 || len > 20 || round(len) != len) {
18. system("cls");
19. printf("Количество элементов массива должно быть целым положительным числом больше одного и не больше двадцати!\n");
20. **continue**;
21. }
22. **break**;
23. }
24. // Проверка на целость - поэтому len типа double, потому что пользователь может ввести дробное число
25. // также идет проверка на введенную единицу в качестве размера алг. прогрессии
26. // количеcтво элементов алг. прогрессии не может быть равным одному

29. printf("Введите значения массива: ");
30. **for**(i = 0; i < len; i++) {
31. scanf("%lg", &numbers[i]);
32. }
34. // сортировка методом пузырька
35. **for**(i = 0; i < len; i++) {
36. **for**(j = 0; j < len - i - 1; j++) {
37. **if** (numbers[j] > numbers[j + 1]) {
38. diff = numbers[j + 1];
39. numbers[j+1] = numbers[j];
40. numbers[j] = diff;
41. }
42. }
43. }
45. // Определение разницы между двумя соседними числами. В арифметической прогрессии
46. // эта разница будет одинакова при любых разностях соседних чисел ариф. прогрес.
47. diff = numbers[1] - numbers[0];
48. **for**(i = 1; i < len - 1; i++) {
49. **if** (fabs((numbers[i + 1] - numbers[i]) - diff) > 0.00001) { // Так как нельзя использовать == с double числами, я применил альтернитиву
50. isTrue = 0; //isTrue - истина, если это ариф. прогрес., ложь, если нет
51. **break**;
52. }
53. }
55. **if**(isTrue) {
56. printf("Введенные элементы являются членами арифметической прогрессии с шагом %lg по модулю\n", diff);
57. } **else** {
58. printf("Это не арифметическая прогрессия\n");
59. }
61. system("pause");
62. **return** 0;
63. }

**Пояснения к программе.**

Код round(len) != len проверяет – является ли число целочисленным.

Ввод элементов осуществлен способ с минимальным количеством итераций

Сортировка пузырьком использует в наихудшем случае N2 итераций, где N – кол-во элементов. Как по мне это не критично, так как у нас может быть всего 20 элементов.

Определение равенства всех разниц соседних элементов происходит за минимальное кол-во итераций, а именно за N-1 итераций, где N – кол-во элементов.

Все операции выполнены с задействием минимально возможного кол-во итераций.

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h.

Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы). А также для очистки консоли (выполняется команда «cls» операционной системы)

Заголовочный файл math.h необходим для использования функции round(), fabs()

**Выводы:**

По результатам лабораторной работы я научился работать с одномерными массивами, организовывать сортировку таковых. Научился за минимальное кол-во итераций заполнять одномерный массив, а также проводить разного рода действия с ним по ходу обхода массива.