**Лабораторная работа №4**

**Полный перебор**

Решить методом полного p-ичного перебора:

Вариант 12

4,12. Арифметическая прогрессия

Дано N (N<=20) натуральных чисел. Вычеркнуть некоторые из них (не более N-3)

так, чтобы из всех оставшихся можно было составить фрагмент арифметической

прогрессии.

Числа записаны в текстовом файле по одному в каждой строке. Выходной файл должен

содержать все возможные варианты прогрессий в отсортированном виде.

Код программы:

#include <iostream>

#include <fstream>

#include <algorithm>

#include <string>

using namespace std;

void checkArProg(int arr[], int n, ofstream& outFile) {

if (n >= 3) {

sort(arr, arr + n);

int diff = arr[1] - arr[0];

if (diff == 0) return;

for (int i = 2; i < n; ++i) {

if (arr[i] - arr[i - 1] != diff) {

return;

}

}

outFile << "Arithmetic progression found: ";

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

outFile << arr[i] << " ";

}

outFile << endl;

}

}

int main() {

ifstream inFile("input.txt");

ofstream outFile("output.txt");

if (!inFile.is\_open()) {

cout << "Unable to open input file";

return 1;

}

const int N = 20;

int n;

inFile >> n;

if (n < 3 || n > 20) {

cout << "Error: n should be between 3 and 20";

return 1;

}

int arr[N];

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

inFile >> arr[i];

if (arr[i] <= 0) {

cout << "Error: numbers should be natural";

return 1;

}

}

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

int arr2[N];

int size = 0;

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

if (mask & (1 << i)) {

arr2[size++] = arr[i];

}

}

checkArProg(arr2, size, outFile);

}

inFile.close();

outFile.close();

ifstream outFile1("output.txt");

string line;

if (outFile1.is\_open()) {

while (getline(outFile1, line)) {

cout << line << endl;

}

outFile1.close();

}

else {

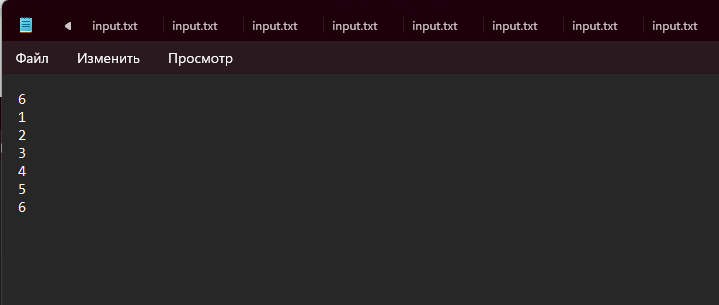
cout << "Unable to open file";

}

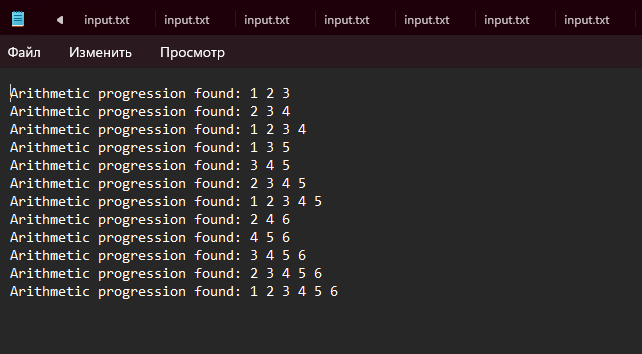
return 0;

}

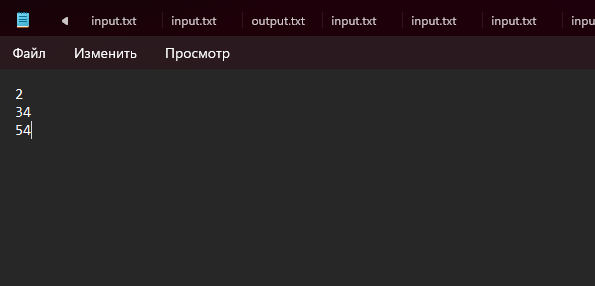
Тесты:

1. input.txt :
2. 

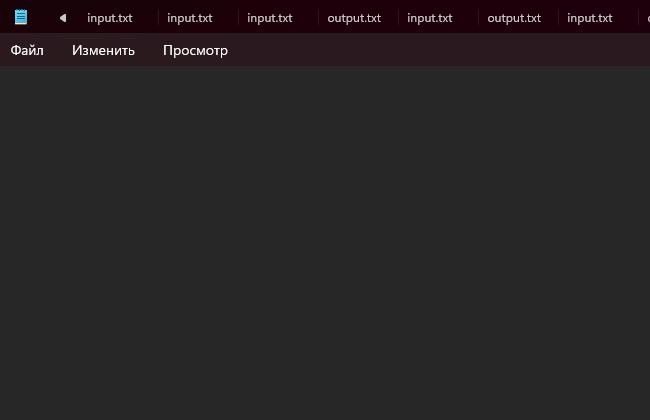
output.txt:



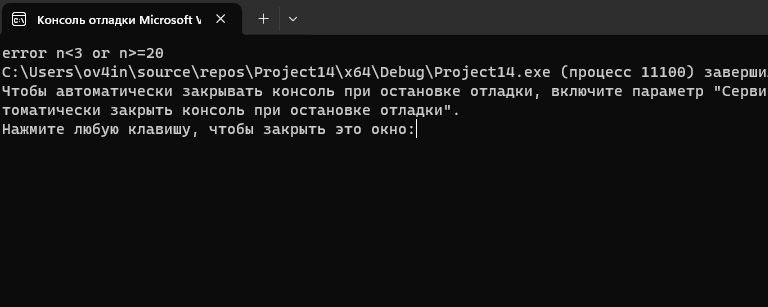
1. input.txt:



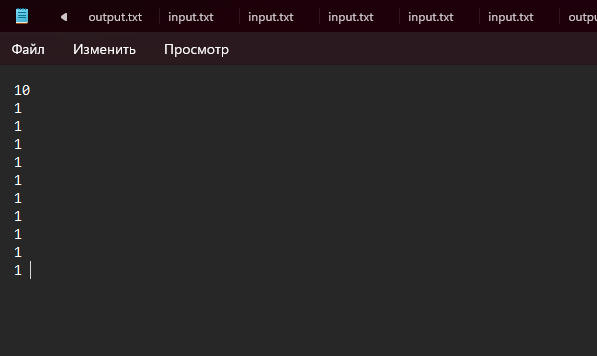
output.txt:



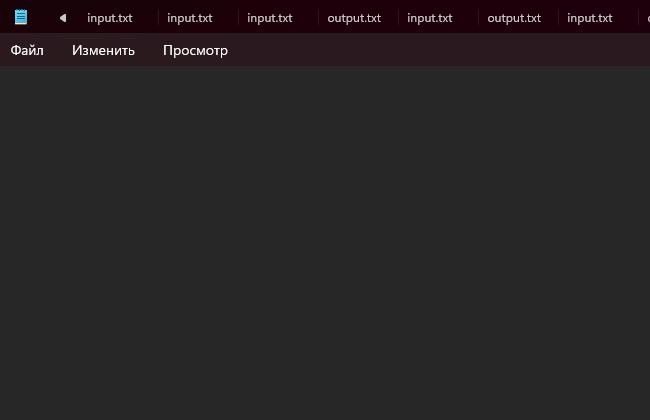
Консоль:



1. input.txt:



output.txt:



Алгоритм:

1. Инициализация генератора случайных чисел с помощью **srand(time(NULL))**.

2. Открытие файла для чтения ввода **(input.txt)** и файла для записи вывода **(output.txt)** с помощью **ifstream** и **ofstream**.

3. Проверка успешного открытия файла ввода. Если не удалось открыть, выводится сообщение об ошибке и программа завершается с кодом ошибки 1.

4. Задание константы **N** со значением 20 и объявление переменных **n**, **arr[N]** и **bin[N]**.

5. Чтение значения **n** из файла ввода.

6. Проверка условия на **n**. Если **n** меньше 3 или больше 20, выводится сообщение об ошибке и программа завершается с кодом ошибки 1.

7. Чтение **n** чисел из файла ввода в массив **arr[]**. Проверяется, что числа являются натуральными (больше нуля).

8. Перебор всех возможных подмножеств:

* Для каждой битовой маски **mask** от 1 до 2^n:
* Создание массива **arr2**, содержащего подмножество чисел из **arr**, соответствующее текущей битовой маске **mask**.
* Вызов функции **checkArProg(arr2, size, outFile**), которая проверяет наличие арифметической прогрессии в подмножестве чисел и записывает результат в файл "output.txt".

9. Закрытие файлов ввода и вывода.

10. Открытие файла вывода для чтения с помощью **ifstream**.

11. Если файл открыт успешно, считывание строк из файла и вывод на экран. Если файл не удалось открыть, выводится сообщение об ошибке.

12. Завершение программы.

Алгоритм функции **checkArProg():**

1. Проверка условия **n >= 3**, если условие не соблюдается, все операции внутри функции не будут работать.
2. Сортировка массива с помощью функции **sort.**
3. Нахождение разницы между 1 и 0 элементом массива.
4. Проверка, если разность равна 0, то выход из функции.
5. Проверка всех оставшихся элементов массива на разность с помощью цикла:

* Если разность между текущем и предыдущем элементом массива не соответствует разности между 1 и 0 элементом, выход из функции.

1. Если массив прошёл проверку удачно, то он образует арифметическую прогрессию и этот массив записывается в **output.txt**