**Лабораторная работа №4**

**Полный перебор**

**Вариант 10**

Задание

Решить методом полного p-ичного перебора следующую задачу:

2,10. Парламент

В государстве Закидонии все важнейшие вопросы решает парламент, состоящий из депутатов. Закон считается принятым, если 51% депутатов проголосовали за него. Однако депутаты не полностью самостоятельны в своем решении: каждый из них входит в какую-то фракцию, а члены фракции должны голосовать одинаково, независимо от личного мнения.

Число депутатов во фракциях различно, поэтому одни из них имеют больший вес, а другие почти не влияют на результат голосования.

Все фракции пронумерованы. Даны число членов всех фракций и номер А некоторой фракции. Необходимо вывести те наборы фракций, для которых голос А будет решающим, т.е. без А не набирается 51% голосов, а с А набирается.

Число фракций - не более 20. Во входном файле указано количество членов каждой фракции по одному числу в строке. Число А (номер фракции) вводится с клавиатуры.

Программный код:

#include <iostream>

#include <fstream>

using namespace std;

void Plus(int\* m, int n) {

int i = n;

while (m[i] == 1) m[i--] = 0;

m[i] = 1;

}

void main()

{

setlocale(LC\_ALL, "ru");

int count = 0; // счетчик количества фракций

int zov; // переменная для временного хранения числа членов фракций, считанного из файла

int sum = 0; // сумма общего числа членов фракций

int fraction; // переменная определяющая, решающую фракцию

ifstream file; // переменная для чтения из файла

file.open("Brute\_force.txt");

if (!file.is\_open())

{

cout << "Ошибка при открытии файла" << endl;

return;

}

cout << "Фракции и количество их участников:" << endl;

// Это конструкция цикла while, который продолжает выполнение своего тела, пока данные могут быть успешно считаны из файла в переменную zov.

while (file >> zov)

{

//++count инкрементирует переменную count, увеличивая её на 1 до её использования в выводе.

cout << ++count << ". " << zov << endl;//Здесь происходит вывод на экран.

}

file.close(); // подсчитывание количество фракций и вывод числа членов фракции с нумерацией каждой

if (count > 20)

{

cout << "Количество фракций больше 20!";

return;

}

file.open("Brute\_force.txt");

// count + 1 используется для создания массива, в котором индексация начинается с 1 и идет до count,

// что соответствует нумерации каждой фракции в задаче о голосовании в парламенте.

int\* arrayFractions = new int[count + 1];// хранит информацию о количестве членов в каждой фракции

arrayFractions[0] = 0;//Устанавливает первый элемент массива в 0.

for (int i = 1; i <= count; i++)

{

file >> arrayFractions[i];

sum += arrayFractions[i];

}

file.close(); // добавление количества членов каждой фракции по отдельности в массив и нахождение суммы всех элементов

cout << endl << "Голосов должно быть больше, чем " << sum / 2 << endl;

cout << endl << "Выберите номер фракции, голос которой будет является решающим: ";

cin >> fraction;

cout << endl << "Комбинации фракций, где голос фракции " << fraction << " является решающим (набирается более 51% голосов)" << endl;

// Размер count + 1 позволяет обрабатывать все возможные комбинации фракций, включая те случаи,

// когда ни одна фракция не участвует в голосовании (binary содержит только 0) или когда участвуют все фракции (binary содержит только 1).

// Это упрощает итерацию по всем возможным комбинациям.

int\* binary = new int[count + 1];

for (int i = 0; i <= count; i++)

{

binary[i] = 0;

}

while (binary[0] == 0)

{

if (binary[fraction] == 0)

{ // беруться только те итерации, где выбранная фракция ещё не проголосовала

int summ = 0; //переменная для подсчета голосов за

for (int i = 1; i <= count; i++)

{

if (binary[i] == 1)

{

summ += arrayFractions[i];

}

}

if ((summ <= (sum / 2)) && (summ + arrayFractions[fraction] > (sum / 2)))

{ // если сумма итерации меньше или равна половине всех голосов и если сумма итерации + голос нашей фракции больше половины всех голосов

for (int i = 1; i <= count; i++)

{

if (binary[i] == 1) {

cout << arrayFractions[i] << " ";

}

}

cout << endl;

}

}

Plus(binary, count); // следующая итерация

}

}

Результаты

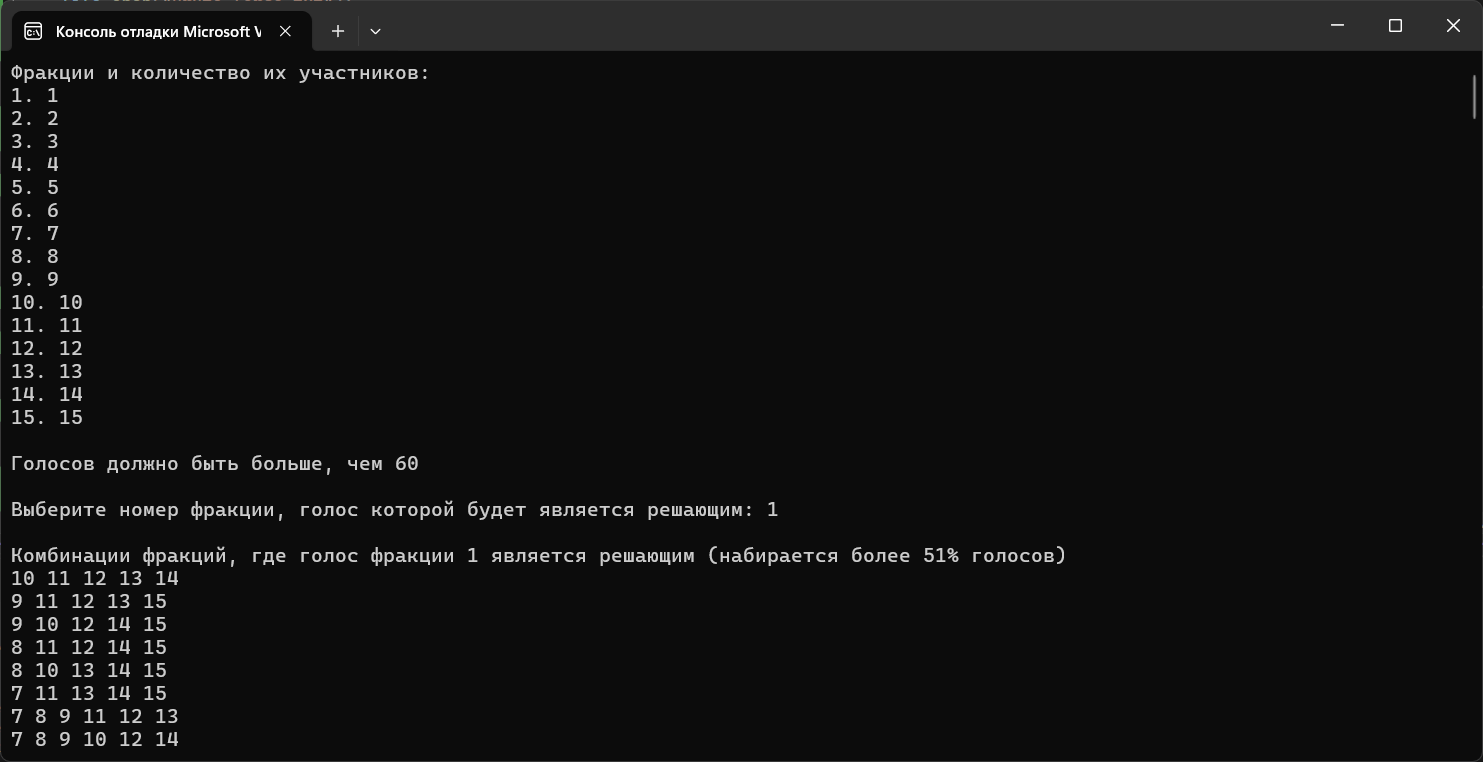
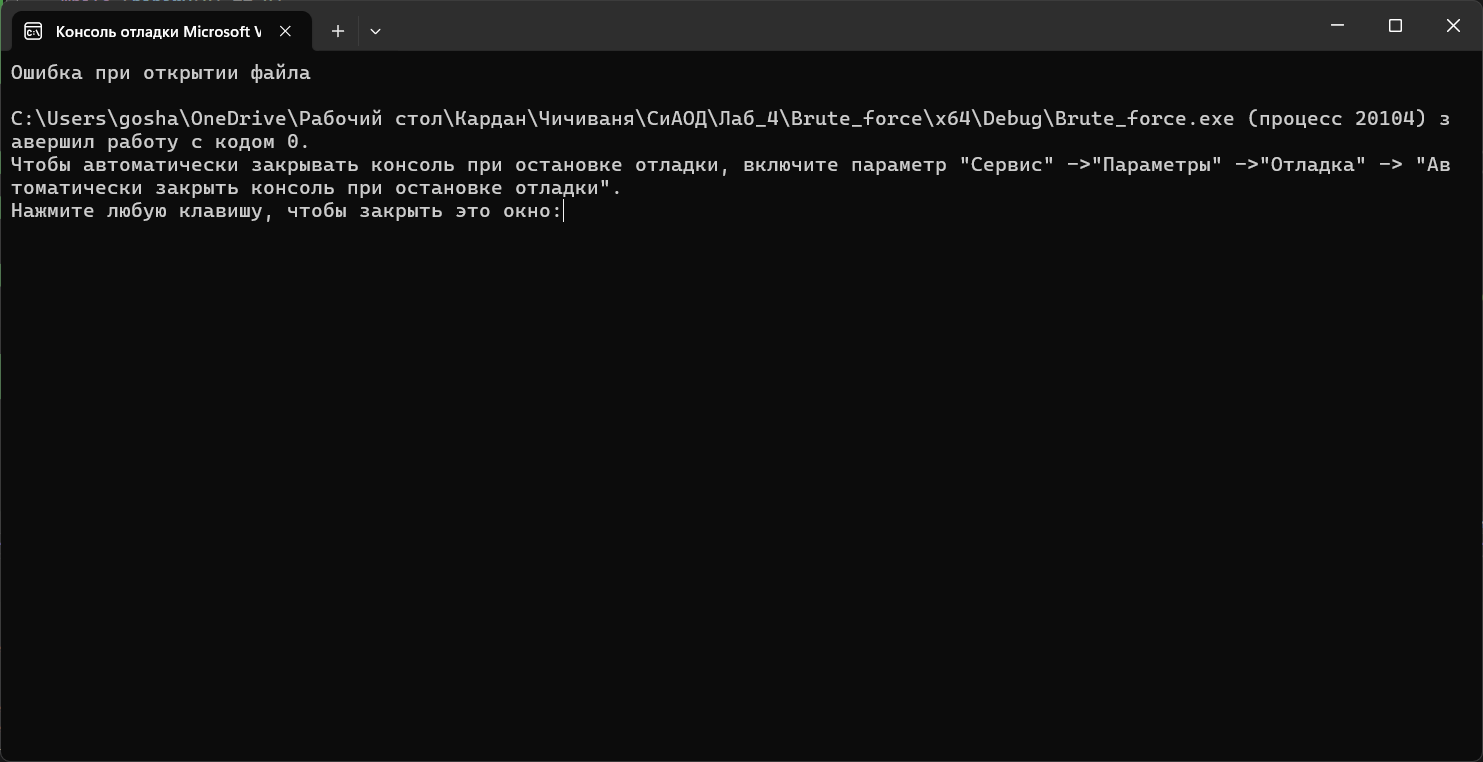


Рис. 1. Вывод программы

  
Рис. 2. Вывод программы, если файл не найден

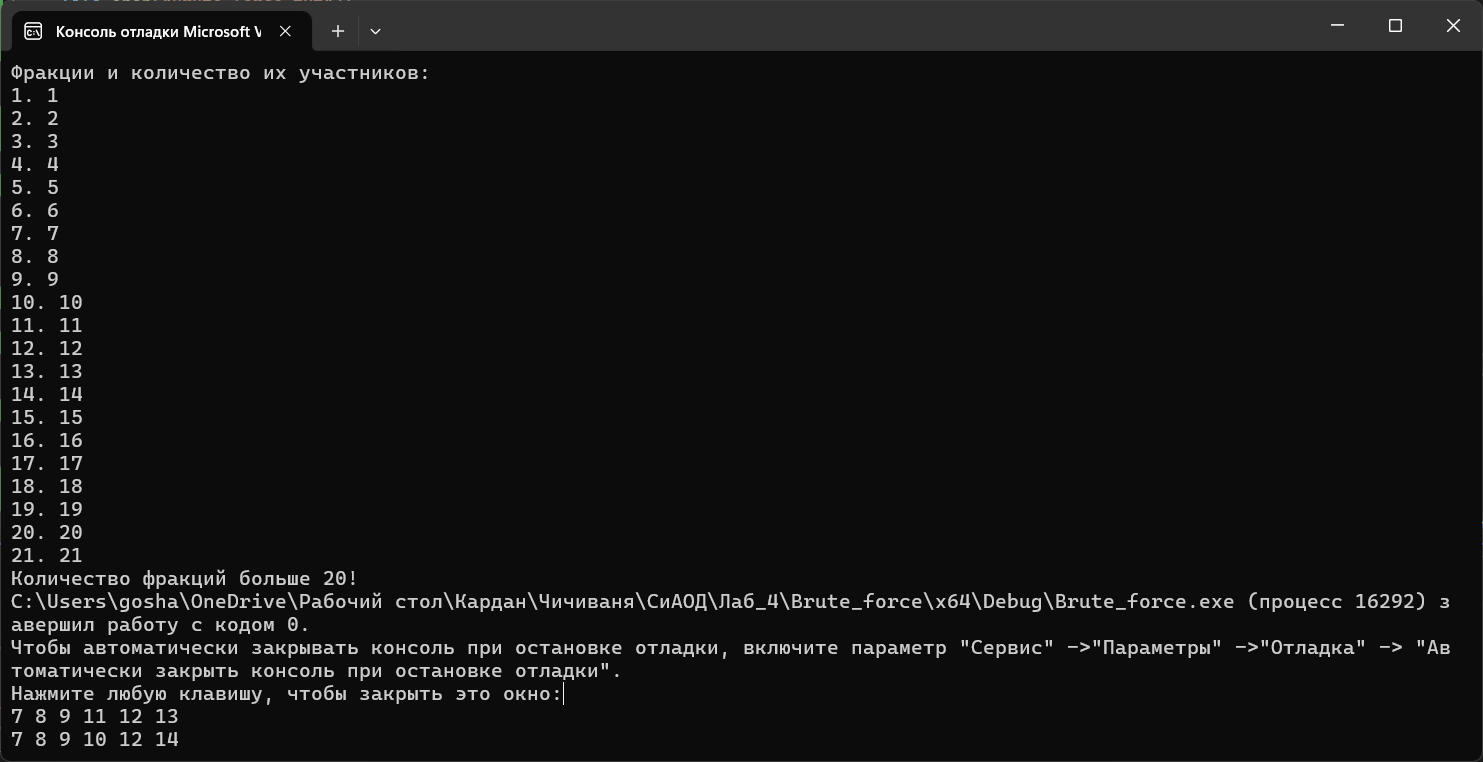


Рис. 3. Вывод программы, фракций больше 20

Описание алгоритма работы программы.

Функция Plus, которая выполняет операцию инкрементирования бинарного числа, представленного массивом m размера n. Эта функция используется для генерации всех возможных комбинаций голосов фракций.

Инициализируются переменные:

* count для подсчета количества фракций
* zov для временного хранения числа членов фракций, считанного из файла
* sum для подсчета суммы количества членов всех фракций
* fraction для определения, решающей фракции
* file определяющая, решающую фракцию

Открывается файл "Brute\_force.txt", содержащий информацию о числе членов каждой фракции. Если файл не удалось открыть, программа выводит сообщение об ошибке и завершает работу.

Далее считываются данные из файла и выводится информация о каждой фракции, включая их номер и количество членов.

Если количество фракций превышает 20, программа выводит сообщение об ошибке и завершает работу.

Снова открывается файл "Brute\_force.txt", и данные о количестве членов каждой фракции по отдельности считываются в массив arrayFractions. Также подсчитывается общее число членов фракций.

Выводится сообщение о том, сколько голосов должно быть в общей сумме для принятия решения.

Пользователю предлагается выбрать номер фракции, голос которой будет решающим.

Выводится сообщение о том, что будут выведены комбинации фракций, где голос выбранной фракции является решающим.

int\* binary = new int[count + 1]; - создается массив binary длиной count + 1, где каждый элемент массива представляет фракцию и может принимать значение 0 или 1. Значение 1 означает, что данная фракция проголосовала за предложение, а 0 - что не проголосовала.

Далее начинается цикл while (binary[0] == 0), который выполняется до тех пор, пока первый элемент массива binary равен 0, что означает, что есть еще комбинации фракций для рассмотрения.

Внутри цикла проверяется условие if (binary[fraction] == 0), что означает, что фракция fraction еще не проголосовала за предложение.

Затем происходит подсчет общего количества голосов за предложение в текущей комбинации фракций.

Если общее количество голосов меньше или равно половине от общего числа депутатов (sum/2), и при этом при добавлении голоса фракции fraction это количество становится больше половины, то эта комбинация фракций выводится на экран как решающая.

В конце выводятся фракции, проголосовавшие за предложение, составляющие решающую комбинацию.

После этого вызывается функция Plus(binary, count), которая увеличивает текущую комбинацию фракций для следующей итерации.