**Лабораторная работа №4**

**Полный перебор**

**14 вариант**

*Решить методом полного p-ичного перебора следующие задачи:*

*Текстовый файл состоит из двух строк, длинами не более N (N<=20). Выяснить,*

*можно ли получить вторую строку из первой путем вычеркивания некоторых символов.*

*Прописные и строчные буквы не различаются, т.е. строки abcd и AbcD считаются одинаковыми.*

*Программа должна находить номера вычеркнутых символов и выводить в текстовый*

*файл все возможные варианты.*

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace Лаб3

{

class Program

{

static List<List<int>> GenerateCombinations(int n)

{

List<List<int>> combinations = new List<List<int>>();

// Генерируем двоичные числа от 0 до 2^n - 1

for (int i = 0; i < Math.Pow(2, n); i++)

{

List<int> combination = new List<int>();

// Перебираем биты числа i

for (int j = 0; j < n; j++)

{

if ((i & (1 << j)) != 0)

{

combination.Add(j);

}

}

combinations.Add(combination);

}

return combinations;

}

static void Main()

{

// Читаем текстовый файл

string[] lines = File.ReadAllLines("input.txt");

if (lines.Length == 0 || lines.Length == 1)

{

Console.WriteLine("Должно быть 2 строки");

return;

}

// Приводим строки к нижнему регистру и удаляем пробелы

string firstLine = lines[0].ToLower().Replace(" ", "");

string secondLine = lines[1].ToLower().Replace(" ", "");

// Если строки равны, выводим это на консоль и завершаем программу

if (firstLine == secondLine)

{

Console.WriteLine("Строки изначально равны");

return;

}

if (firstLine.Length < secondLine.Length)

{

Console.WriteLine("Невозможно привести к одинаковому виду");

}

int n = firstLine.Length;

if (n > 20)

{

Console.WriteLine("Ошибка выполнения, элементов должно быть меньше или равно 20");

}

else

{

// Генерируем все возможные комбинации вычеркнутых символов

List<List<int>> combinations = GenerateCombinations(n);

// Проверяем каждую комбинацию

using (StreamWriter file = new StreamWriter("output.txt"))

{

foreach (List<int> combination in combinations)

{

string modifiedLine = firstLine;

// Вычеркиваем символы по номерам из комбинации

foreach (int index in combination.OrderByDescending(i => i))

{

modifiedLine = modifiedLine.Remove(index, 1);

}

// Если получили вторую строку, записываем комбинацию в текстовый файл

if (modifiedLine == secondLine)

{

// Добавляем +1 к каждому элементу комбинации перед записью в файл

List<string> adjustedCombination = combination.Select(x => (x + 1).ToString()).ToList();

file.WriteLine(string.Join(" ", adjustedCombination));

}

}

}

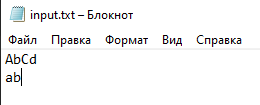
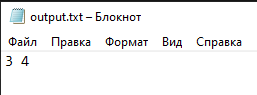
// Генерация всех возможных комбинаций вычеркнутых символов

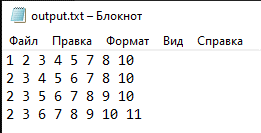
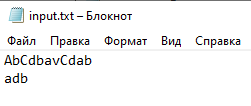
}

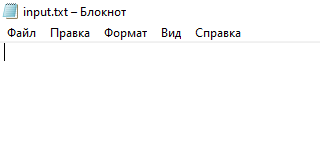
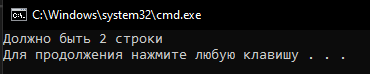
}

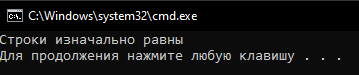
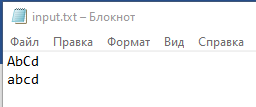
}

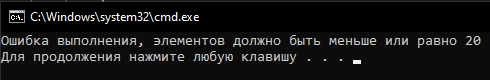
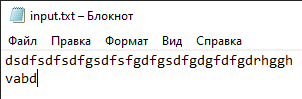
}

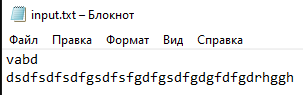
 

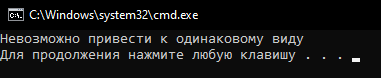










Алгоритм программы:

* Чтение строк из файла “input.txt” и **помещение в массив**
* Проверка на наличие строк в файле, если не выполняется условие, то выводится сообщение об ошибке
* Приводим строки к нижнему регистру
* Если строки изначально равны, то выводим сообщение об этом и завершаем программу
* Если длина первой строки будет меньше длины второй строки, то выводим об этом сообщение и завершаем программу
* Если длина первой строки > 20, то выводим сообщение об этом и завершаем программу
* Иначе начинаем генерировать все возможные комбинации вычеркнутых символов из первой строки путем вызова метода GenerateCombinations. В методе мы создаем двойной список/список в списке, который будет содержать все сгенерированные комбинации. Через цикл for мы генерируем двоичные представление чисел от 0 до 2^n - 1. Для каждого i в цикле создается новый список, представляющий текущую комбинацию. Далее организуется еще один цикл for от 0 до n-1 для перебора битов числа i. Внутри этого цикла проверяется, установлен ли j-й бит в числе i при помощи поразрядной операции "И", сдвигая 1 влево на j позиций и применяя поразрядное "И" с числом i. Если проверка возвращает ненулевое значение, то j-й бит установлен, и это число включается в текущую комбинацию. По завершению цикла текущая комбинация добавляется в изначальный список. Метод возвращает список combinations, содержащий все сгенерированные комбинации.
* Далее берем комбинации вычеркнутых символов, сортируем комбинация в порядке убывания, чтобы удаление происходило справа налево, чтобы не портить порядок следования символов в строке. На каждой итерации удаляется символ из строки modifiedLine на позиции index. Сравниваем полученные строки, если совпадает, то комбинация номеров вычеркнутых элементов записывается в файл output.txt