Лабораторная работа №4

Полный перебор

Решить методом полного p-ичного перебора следующие задачи:

Вариант 6: 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.

Результат работы программы представлен в соответствии с рис.1-6:

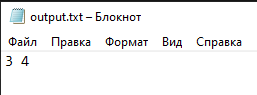
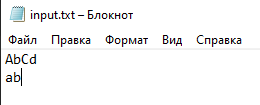


Рис. 1.

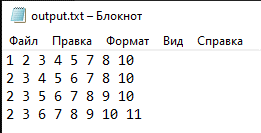
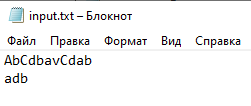


Рис. 2.

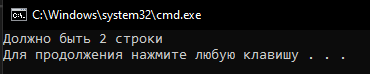
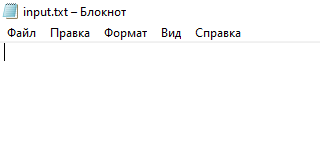


Рис. 3.

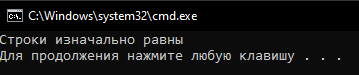
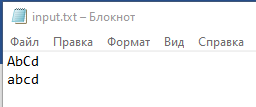


Рис. 4.

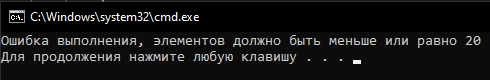
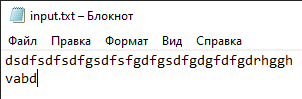


Рис. 5.

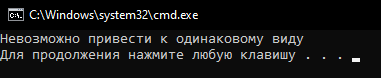
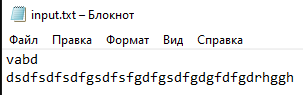


Рис. 6.

Алгоритм работы программы:

1. Чтение строк из файла "input.txt" и помещение их в переменные firstLine и secondLine. Проверка на наличие строк в файле: если хотя бы одна строка отсутствует, выводится сообщение об ошибке и программа завершается.
2. Приведение обеих строк к нижнему регистру с помощью функции ToLowerCase, которая проходит по каждому символу строки и изменяет заглавные буквы на строчные.
3. Удаление всех пробелов из обеих строк с помощью функции RemoveSpaces, которая создает новую строку и добавляет все символы, кроме пробелов.
4. (Проверка на равенство) Сравниваются две строки. Если они изначально равны, выводится сообщение об этом и программа завершается.
5. Проверка длины строк:

Если длина первой строки меньше длины второй, выводится сообщение о невозможности приведения строк к одинаковому виду, и программа завершается.

Если длина первой строки превышает 20 символов, выводится сообщение об ошибке и программа завершается.

1. Начинается процесс генерации всех возможных комбинаций вычеркнутых символов из первой строки путем вызова функции GenerateCombinations.

Внутри функции вычисляется общее количество комбинаций, равное 0 до 2^n - 1, где n – длина первой строки.

Затем в цикле от 0 до (2^n – 1) перебираются все возможные маски. Для каждой маски создается копия первой строки (modifiedLine), в которую будут вноситься изменения.

Создается массив indices для хранения индексов символов, которые будут вычеркнуты, и переменная count для отслеживания количества вычеркнутых символов.

Внутри вложенного цикла проверяются биты текущего индекса. Если установленный бит, сохраняется его индекс.

После завершения цикла по битам осуществляется удаление символов в modifiedLine по индексам из массива indices. Удаление происходит в обратном порядке, чтобы избежать ошибок индексации.

После удаления символов происходит сравнение первой строки со второй строкой. Если они совпадают, индексы вычеркнутых символов записываются в файл output.txt с прибавлением 1 для корректной нумерации.

Код программы:

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

using namespace std;

// Функция для преобразования строки к нижнему регистру

void ToLowerCase(string& str) {

for (size\_t i = 0; i < str.length(); i++) {

if (str[i] >= 'A' && str[i] <= 'Z') {

str[i] = str[i] + ('a' - 'A');

}

}

}

// Функция для удаления пробелов из строки

void RemoveSpaces(string& str) {

string result;

for (char c : str) {

if (c != ' ') {

result += c;

}

}

str = result;

}

// Функция для генерации всех возможных комбинаций

void GenerateCombinations(int n, string firstLine, string secondLine, ofstream& outputFile) {

int totalCombinations = pow(2, n); // Общее количество комбинаций

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

string modifiedLine = firstLine;

int indices[20]; // Массив для хранения индексов символов, которые будут вычеркнуты

int count = 0;

// Перебираем биты числа i

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

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

indices[count++] = j; // Сохраняем индекс вычеркиваемого символа

}

}

// Вычеркиваем символы по номерам из массива индексов, перебирая его в обратном порядке

for (int j = count - 1; j >= 0; j--) {

modifiedLine.erase(modifiedLine.begin() + indices[j]);

}

// Если полученная строка равна второй строке, записываем комбинацию в текстовый файл

if (modifiedLine == secondLine) {

for (int j = 0; j < count; j++) {

outputFile << (indices[j] + 1) << " "; // +1 для записи номера символа

}

outputFile << endl;

}

}

}

int main() {

setlocale(LC\_ALL, "rus");

// Чтение текстового файла

ifstream inputFile("input.txt");

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

cout << "Не удается открыть файл input.txt" << endl;

return 1;

}

string firstLine, secondLine;

getline(inputFile, firstLine);

getline(inputFile, secondLine);

inputFile.close();

if (firstLine.empty() || secondLine.empty()) {

cout << "Должно быть 2 строки" << endl;

return 1;

}

// Приведение строк к нижнему регистру и удаление пробелов

ToLowerCase(firstLine);

RemoveSpaces(firstLine);

ToLowerCase(secondLine);

RemoveSpaces(secondLine);

// Проверка на равенство строк

if (firstLine == secondLine) {

cout << "Строки изначально равны" << endl;

return 0;

}

if (firstLine.length() < secondLine.length()) {

cout << "Невозможно привести к одинаковому виду" << endl;

return 1;

}

int n = firstLine.length();

if (n > 20) {

cout << "Ошибка выполнения, элементов должно быть меньше или равно 20" << endl;

return 1;

}

// Генерация всех возможных комбинаций вычеркнутых символов

ofstream outputFile("output.txt");

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

cout << "Не удается открыть файл output.txt" << endl;

return 1;

}

GenerateCombinations(n, firstLine, secondLine, outputFile);

outputFile.close();

return 0;

}