**Отчёт по лабораторной работе №4**

**Полный перебор**

Задание:

Решить методом полного p-ичного перебора следующую задачу:

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.

Код программы:

|  |
| --- |
| main.py |
| file = open("file.txt", "r").readlines()  output = open("output.txt", "w")  file[0] = file[0].lower().strip()  file[1] = file[1].lower().strip()  founded = False  for i in range(1 << len(file[0])):  bin\_str = f"{i:0{len(file[0])}b}"  indexes = [ i for i in range(len(bin\_str)) if bin\_str[i] == "1" ]  new\_str = file[0]  for j in range(len(indexes)):  new\_str = new\_str[:indexes[j]-j] + new\_str[indexes[j]-j+1:]  if new\_str == file[1]:  output.write(str(indexes)+"\n")  founded = True  if not founded: output.write("Not possible!")  output.close() |

Алгоритм работы:

Программа читает две строки из файла, также при этом открывает для записи файл, в который позже будет записан результат. Затем происходит вхождение в цикл, который работает количество раз, равное двоичному числу, количество разрядов которого равно длине первой строки и все разряды - единицы. Достигается это число при помощи побитового сдвига единицы влево на количество разрядов равное количеству строки (на выходе получится, например, 100000000), а затем вычитанием из полученного числа единицы (на выходе получится 11111111). Внутри цикла номер текущего прохода цикла преобразовывается в двоичную строку длиной равной первой строке из файла. Эта строка является представлением какой символ первой строки из файла вычёркивается (1), а какой нет (0). Далее исходя из вышеописанной строки формируется массив индексов, на которых в строке находятся единицы. Затем формируется временная строка, из которой по циклу отнимаются символы, стоящие на найденных индексах. После полученная временная строка сравнивается со второй строкой из файла, и, если строки совпадают – в выходной файл записывается массив индексов «вычеркнутых» символов. Если по выходу из основного цикла совпадающих строк найдено не было – в выходной файл выводится сообщение о невозможности решения задачи. В конце работы программы файл закрывается.

Результат работы:

* Тест 1:

Строка 1 – kduhhytffffcvggbgj

Строка 2 – kuytfcvbj

Результат:

[1, 3, 4, 8, 9, 10, 13, 14, 16]  
[1, 3, 4, 7, 9, 10, 13, 14, 16]  
[1, 3, 4, 7, 8, 10, 13, 14, 16]  
[1, 3, 4, 7, 8, 9, 13, 14, 16]

* Тест 2:

Строка 1 – agbcqrdfdehsfghdijskh

Строка 2 – abcdefghijkh

Результат:

[1, 4, 5, 7, 8, 10, 11, 15, 18]  
[1, 4, 5, 6, 7, 10, 11, 15, 18]

* Тест 3:

Строка 1 – giycviahsdbivkcer

Строка 2 – wdhcjwafw

Результат:

Not possible!

* Тест 4:

Строка 1 – 0978095712348868i5

Строка 2 – 12345

Результат:

[0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 15, 16]

* Тест 5:

Строка 1 – Coompplete z-point seaarch

Строка 2 – Complete p-point search

Результат:

Not possible!