using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace ConsoleApplication1

{

class Program

{

public static void ShowAllCombinations<T>(IList<T> arr, string current = "")

{

if (arr.Count == 0) //если все элементы использованы, выводим на консоль получившуюся строку и возвращаемся

{

Console.WriteLine(current);

return;

}

for (int i = 0; i < arr.Count; i++) //в цикле для каждого элемента прибавляем его к итоговой строке, создаем новый список из оставшихся элементов, и вызываем эту же функцию рекурсивно с новыми параметрами.

{

List<T> lst = new List<T>(arr);

lst.RemoveAt(i);

ShowAllCombinations(lst, current + arr[i].ToString());

}

}

static void Main(string[] args)

{

/\*

Задание 1.

Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами.

В каждом слове заменить первую букву на прописную.

\*/

Console.WriteLine("Задание 1.");

Console.WriteLine("Введите строку: ");

string str1 = Console.ReadLine();

char[] arr = str1.ToCharArray();

arr[0] = char.ToUpper(arr[0]);

for (int i = 1; i < str1.Length; i++)

{

if (arr[i - 1] == ' ') arr[i] = char.ToUpper(arr[i]);

}

str1 = new string(arr);

Console.WriteLine(str1);

Console.WriteLine();

/\*

Задание 2-3.

1. Дана строка. Определить, сколько раз входит в нее группа букв (abc).

2. Дана строка. Преобразовать ее, заменив точками все двоеточия,

встречающиеся среди первой половины символов строки, и заменив точками все восклицательные.

\*/

Console.WriteLine("Задание 2-3.");

Console.WriteLine("Введите строку: ");

string str2 = Console.ReadLine();

Regex reg = new Regex("abc");

MatchCollection math = reg.Matches(str2);

Console.WriteLine("Нашел {0} совпадений!", math.Count);

double temp = Convert.ToDouble(str2.Length) / 2;

char[] chars = str2.ToCharArray();

for (int i = 0; i < str2.Length; i++)

{

if (str2[i] == '.')

chars[i] = '!';

if (str2[i] == ':' && i < temp)

chars[i] = '.';

}

str2 = new string (chars);

Console.WriteLine(str2);

Console.WriteLine();

/\*

Задание 4.

Составить программу, которая будет вводить строку в переменную string.

Удалить из нее все лишние пробелы, оставив между словами не более одного.

Результат поместить в новую строку.

\*/

Console.WriteLine("Задание 4.");

Console.WriteLine("Введите строку: ");

string str4 = Console.ReadLine();

str4 = string.Join(" ", str4.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries));

Console.WriteLine(str4);

Console.WriteLine();

/\*

\* Задание 5.

\* 9. C клавиатуры вводится предложение, слова в котором разделены символом ‘\_’. Используя треугольник Паскаля зашифруйте исходное предложение по правилу:

 Из предложения выделяется слово;

 Из треугольника Паскаля выбирается строка с номером равным числу букв в слове;

 К-я буква исходного слова заменяется на букву, отстоящую от исходной на число букв, указанное в к-м столбце выбранной строки треугольника.

1 А Б В Г Д Е Ж З

1 1 Я И

1 2 1 Ю Й

1 3 3 1 Э К

1 4 6 4 1 Ь Л

Ы М

Ъ Н

Щ О

Ш П

Ч Ц Х Ф У Т С Р

\*/

Console.WriteLine("Задание 5.");

// Подготовка данных

// Факториал

const int MAX = 21;

long[] fact = new long[MAX];

fact[0] = fact[1] = 1;

for (int i = 2; i < fact.Length; i++)

{

fact[i] = checked(i \* fact[i - 1]);

}

// Треугольник Паскаля

var pascalTriangle = new int[MAX][];

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

{

pascalTriangle[n] = new int[n + 1];

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

{

checked

{

long combination = fact[n] / (fact[k] \* fact[n - k]);

pascalTriangle[n][k] = (int)combination;

}

}

}

// Шифрование

const string ABC = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

Console.WriteLine("Введите строку: ");

string text = Console.ReadLine();

Console.WriteLine(text);

var sb = new StringBuilder(text.Length);

foreach (string oneWord in text.Split('\_'))

{

int[] pascalRow = pascalTriangle[oneWord.Length - 1];

if (sb.Length > 0) sb.Append("\_");

for (int i = 0; i < oneWord.Length; i++)

{

char ch = oneWord[i];

int idx = ABC.IndexOf(ch);

ch = ABC[(idx + pascalRow[i]) % ABC.Length];

sb.Append(ch);

}

}

text = sb.ToString();

Console.WriteLine(text);

Console.ReadKey();

}

}

}