Лабораторная работа 2 по С#

Работа со строками. Регулярные выражения. Дата и время

Задание 1

//Console.WriteLine("удалить из строки все английские строчные буквы");

//string str20 = "опва\*рррорыfdhgjfhdроfавfkjhgfdfgd лао djkавп fgjfkd\*h";

//Console.WriteLine(str20);

//Console.WriteLine(m20(str20));

//var pare = m12(str20, '\*');

//Console.WriteLine(pare);

//Console.ReadKey();

//string m20(string s)

//{

// for(char c = 'a'; c <= 'z'; c++)

// {

// s = s.Replace(c.ToString(), "");

// }

// return s;

//}

//(int,int) m12(string s, char c)

//{

// return (s.IndexOf(c),s.LastIndexOf(c));

//}

//Вариант 1 определить, какой из двух заданных

//символов встречается чаще в строке;

//char m1(string s, char c1, char c2)

//{

// int k1 = 0; int k2=0;

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

// {

// if (s[i] == c1) k1++;

// if (s[i] == c2) k2++;

// }

// if (k1 > k2) return c1;

// else if (k1<k2) return c2;

// else return '0';

//}

//Console.WriteLine(m1(str20,'j','f'));

//Вариант 2 подсчитать общее число вхождений символов х и y;

//int m2(string s, char c1, char c2)

//{

// int k1 = 0;

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

// {

// if (s[i] == c1 || s[i] == c2) k1++;

// }

// return k1;

//}

//Console.WriteLine(m2(str20, 'j', 'f'));

//Вариант 3 подсчитать количество букв в строке;

//int m3(string s)

//{

// int k1 = 0;

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

// {

// for(char c = 'a'; c <= 'z'; c++)

// {

// if (c == s[i]) k1++;

// }

// for (char c = 'A'; c <= 'Z'; c++)

// {

// if (c == s[i]) k1++;

// }

// for (char c = 'а'; c <= 'я'; c++)

// {

// if (c == s[i]) k1++;

// }

// for (char c = 'А'; c <= 'Я'; c++)

// {

// if (c == s[i]) k1++;

// }

// }

// return k1;

//}

//Console.WriteLine(m3(str20));

//Вариант 4 определить, имеются ли в строке

//два соседствующих одинаковых символа;

//bool m4(string s)

//{

// bool k1 = false;

// for (int i = 0; i < s.Length-1; i++)

// {

// if (s[i] == s[i + 1]) { k1 = true; break; }

// }

// return k1;

//}

//Console.WriteLine(m4(str20));

//Вариант 5 удалить среднюю букву, если длина строки нечетная,

//и две средних, если длина строки четная;

//if (str20.Length % 2 != 0)

//{

// str20 = str20.Remove(str20.Length / 2, 1);

//}

//else if (str20.Length % 2 == 0)

//{

// str20 = str20.Remove(str20.Length / 2, 2);

//}

//Console.WriteLine(str20);

//Вариант 6 удвоить каждое вхождение заданного символа x;

//string m6(string s, char c)

//{

// string g = s.Replace(c.ToString(), c.ToString() + c.ToString());

// return g;

//}

//Console.WriteLine(m6(str20, 'f'));

//Вариант 7 удалить все цифры;

//string str7 = "fdgdhd 1223 rtfdh 21313 ghfdg";

//string m7(string s)

//{

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

// {

// for (int j = 0; j < 10; j++)

// {

// s = s.Replace(j.ToString(), "");

// }

// }

// return s;

//}

//Console.WriteLine(m7(str7));

//Вариант 8 удалить все подстроки substr;

//string str8 = "dan 32423 sjsjks dan sdlkfldan 23323 dan";

//string str = "dan";

//while (str8.Contains(str))

//{

// str8=str8.Remove(str8.IndexOf(str), str.Length);

//}

//Console.WriteLine(str8);

//Вариант 9 заменить все вхождения подстроки substr1 на подстроку substr2;

//string str8 = "dan 32423 sjsjks dan sdlkfldan 23323 dan";

//string str1 = "dan";

//string str2 = "jdkhd";

//string m9(string s, string s1, string s2){

// while (s.Contains(s1))

// {

// s = s.Replace(s1, s2);

// }

// return s;

//}

//Console.WriteLine(m9(str8, str1,str2));

//Вариант 10 подсчитать сумму всех содержащихся в строке цифр;

//string str = "dfjdjf 1234 kfdjgd 12345 kjkdf 1234";

//int m10(string s)

//{

// int sum = 0;

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

// {

// for (int j = 0; j <= 9; j++)

// {

// if (j == s[i]-'0') sum += j;

// }

// }

// return sum;

//}

//Console.WriteLine(m10(str));

//Вариант 11 подсчитать количество содержащихся в строке цифр;

//string str = "dfjdjf 1234 kfdjgd 12345 kjkdf 1234";

//int m11(string s)

//{

// int k = 0;

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

// {

// for (int j = 0; j <= 9; j++)

// {

// if (j == s[i] - '0') k++;

// }

// }

// return k;

//}

//Console.WriteLine(m11(str));

//Вариант 13 заменить все группы стоящих рядом точек на многоточие;

//string str = "hgfdhgd....... sdfhsdhf.. djhfdjshf....dsjfg......";

//Console.WriteLine(str);

//string m13(string s)

//{

// string str2 = "...";

// while (s.Contains(".."))

// {

// bool fl= false;

// int n = 0;

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

// {

// if (s[i] == '.') { fl = true; n++; }

// else if (s[i] != '.') { fl = false; n= 0; }

// if (n > 1 && fl ==true) { s = s.Remove(i, 1); }

// }

// }

// return s;

//}

//Console.WriteLine(m13(str));

//Вариант 14 вывести на экран последовательность символов,

//расположенных до первого двоеточия;

//string str= "dksfghfdhjjsidj:sigisgisjijs";

//string m14(string s)

//{

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

// {

// if(s[i]==':') s=s.Remove(i);

// }

// return s;

//}

//Console.WriteLine(m14(str));

//Вариант 15 вывести на экран последовательность символов,

//расположенных после последнего двоеточия;

//string str = "dksfghf:dhjjsidj:sigisg:isjijs";

//string m15(string s)

//{

// s = s.Remove(0,s.LastIndexOf(":")+1);

// return s;

//}

//Console.WriteLine(m15(str));

//Вариант 16 удалить из строки последовательность символов,

//расположенных между круглыми скобками(считается,

//что в строке ровно одна пара круглых скобок).

//string str = "ghksdgks(sgsjgskskgjks)sfjgkdjkgd";

//string m16(string s)

//{

// s=s.Remove(s.IndexOf("("), s.IndexOf(")")- s.IndexOf("(")+1);

// return s;

//}

//Console.WriteLine(m16(str));

//Вариант 17 удалить из строки последовательность символов,

//расположенных между двумя запятыми(считается,

//что в строке ровно две запятые);

//string str = "ghksdgks,sgsjgskskgjks,sfjgkdjkgd";

//string m17(string s)

//{

// s = s.Remove(s.IndexOf(","), s.LastIndexOf(",") - s.IndexOf(",") + 1);

// return s;

//}

//Console.WriteLine(m17(str));

//Вариант 18 определить, сколько различных символов встречается в строке;

//string str = "zshgjdgkmskbninieqjfijnsdkdgjsij";

//int m18(string s)

//{

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

// {

// for(int j = s.IndexOf(s[i])+1; j < s.Length; j++)

// {

// if (s[j] == s[i]) s = s.Remove(j, 1);

// }

// }

// int n = 0;

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

// {

// n++;

// }

// return n;

//}

//Console.WriteLine(m18(str));

//Вариант 19 заменить все цифры в строке на \*;

//string str = "dsuhshsdfsj287587zxhcj38475zxncj23472";

//Console.WriteLine(str);

//string m19(string s)

//{

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

// {

// int n = 0;

// for (int j = 0; j < 10; j++)

// {

// if (j == s[i] - '0')

// {

// s = s.Remove(i, 1);

// s = s.Insert(i, "\*");

// }

// }

// }

// return s;

//}

//Console.WriteLine(m19(str));

Задание 2

using System;

using System.Text;

using System.Threading.Tasks.Dataflow;

//Console.WriteLine("закодировать строку, используя алгоритм Цезаря(только англ буквы)");

StringBuilder sb;

StringBuilder sb1 = new StringBuilder("H,ellfo,world!");

StringBuilder sb2 = new StringBuilder("При(веfт123)4..... в,се,м!");

StringBuilder sb3 = new StringBuilder("Абраf(када)бра.... \*\*\*\*");

StringBuilder sb4 = new StringBuilder("(12)34 ,1234, 567...... 890");

//Console.WriteLine(m2(out sb, sb1, sb2, sb3, sb4));

//Console.WriteLine(sb);

//int m2(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for(int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] >= 'a' && args[i][j] <= 'z'||

// args[i][j] >= 'A' && args[i][j] <= 'Z')

// {

// args[i][j]=(char)((args[i][j] - 'a' + 13) % 26 + 'a');

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// }

// return result.EnsureCapacity(result.Length);

//}

//Вариант 1 поменять местами соседние символы в каждой паре символов

//StringBuilder res;

//StringBuilder m1(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for (int j = 0; j < args[i].Length; j+=2)

// {

// char c = args[i][j];

// args[i][j] = args[i][j+1];

// args[i][j+1] = c;

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m1(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 3 удалить среднюю букву, если длина строки нечетная,

//и две средних, если длина строки четная;

//StringBuilder res = new StringBuilder();

//StringBuilder m3(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// if (args[i].Length%2 == 0)

// {

// args[i] = args[i].Remove(args[i].Length/2, 2);

// }

// else

// {

// args[i] = args[i].Remove(args[i].Length / 2, 1);

// }

// }

// result =new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m3(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 5 заменить все цифры в строке на \*

//StringBuilder res = new StringBuilder();

//StringBuilder m5(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for (int j = 0; j < args[i].Length; j++)

// {

// for (int k = 0; k < 10; k++)

// {

// if (k == args[i][j] - '0') { args[i][j] = '\*'; }

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m5(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 6 удалить все подстроки substr;

//StringBuilder res = new StringBuilder();

//StringBuilder m6(out StringBuilder result, string str, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// args[i] = args[i].Replace(str, "");

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//string str = "1234";

//m6(out res,str, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 7 заменить все вхождения подстроки substr1 на подстроку substr2;

//StringBuilder res = new StringBuilder();

//StringBuilder m7(out StringBuilder result, string str, string str1, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// args[i] = args[i].Replace(str, str1);

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//string str = "1234";

//string str1 = "4321";

//m7(out res, str, str1, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 8 заменить все круглые скобки на квадратные

//StringBuilder res = new StringBuilder();

//StringBuilder m8(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// args[i] = args[i].Replace("(", "[");

// args[i] = args[i].Replace(")", "]");

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m8(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 9 удалить из строки последовательность символов,

//расположенных между круглыми скобками(считается, что в строке ровно одна пара круглых скобок).

//StringBuilder res = new StringBuilder();

//StringBuilder m9(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int a=0;

// int b=0;

// bool fl = false;

// for(int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == '(')

// {

// a = j;

// fl = true;break;

// }

// }

// for (int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == ')')

// {

// b = j;

// }

// }

// if (fl)

// {

// args[i] = args[i].Remove(a, b - a + 1);

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m9(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 10 удалить из строки последовательность символов,

//расположенных между двумя запятыми(считается,

//что в строке ровно две запятые);

//StringBuilder res = new StringBuilder();

//StringBuilder m10(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int a = 0;

// int b = 0;

// bool fl = false;

// for (int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == ',')

// {

// a = j;

// fl = true; break;

// }

// }

// for (int j = a+1; j < args[i].Length; j++)

// {

// if (args[i][j] == ',')

// {

// b = j;

// }

// }

// if (fl)

// {

// args[i] = args[i].Remove(a, b - a + 1);

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m10(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 11 удалить среднюю букву, если длина строки нечетная,

//и две средних, если длина строки четная;

//StringBuilder res = new StringBuilder();

//StringBuilder m11(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// if (args[i].Length%2 == 1)

// {

// args[i] = args[i].Remove(args[i].Length/2, 1);

// }

// if (args[i].Length % 2 == 0)

// {

// args[i] = args[i].Remove(args[i].Length / 2, 2);

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m11(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 12 удвоить каждое вхождение заданного символа x;

//StringBuilder res = new StringBuilder();

//StringBuilder m12(out StringBuilder result, char t, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for(int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == t)

// {

// args[i] = args[i].Insert(j+1,t);j++;

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//char t = 'f';

//m12(out res, t, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 13 заменить все группы стоящих рядом точек на многоточие;

//StringBuilder res = new StringBuilder();

//StringBuilder m13(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int n = 0;

// for (int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == '.')

// {

// n++;

// }

// if(n>1)

// {

// args[i] = args[i].Remove(j, 1);j--;

// }

// }

// args[i] = args[i].Replace(".", "...");

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m13(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 14 в начало строки добавить ее длину

//StringBuilder res = new StringBuilder();

//StringBuilder m14(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int n = args[i].Length;

// args[i] = args[i].Insert(0, n);

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m14(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 15 удалить все знаки препинания

//StringBuilder res = new StringBuilder();

//StringBuilder m15(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for(int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] =='.' || args[i][j] ==',' || args[i][j] ==':' || args[i][j] == ';' || args[i][j] == '!')

// {

// args[i] = args[i].Remove(j, 1);j--;

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m15(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 16 удалить все пробелы

//StringBuilder res = new StringBuilder();

//StringBuilder m16(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for (int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == ' ')

// {

// args[i] = args[i].Remove(j, 1); j--;

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m16(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 17 считая, что в строке содержится фамилия,

//имя и отчество, преобразовать в инициалы(Пушкин Александр

//Сергеевич->Пушкин А.С.)

//StringBuilder sb1 = new StringBuilder("Пушкин Александр Сергеевич");

//StringBuilder sb2 = new StringBuilder("Василевский Евгений Валерьевич");

//StringBuilder sb3 = new StringBuilder("Абражевич Александр Владимирович");

//StringBuilder sb4 = new StringBuilder("Розов Никита Сергеевич");

//StringBuilder res = new StringBuilder();

//StringBuilder m17(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int n = 0;

// for (int j = 0; j < args[i].Length; j++)

// {

// if (args[i][j] == ' ')

// {

// n++;

// }

// if (n > 0)

// {

// for(char k='а'; k<='я'; k++)

// {

// if(args[i][j]==k)

// {

// args[i] = args[i].Remove(j, 1);j--;

// }

// }

// }

// if (n == 2 && args[i][j]==' ') args[i] = args[i].Remove(j, 1);

// for(char k = 'А'; k < 'Я'; k++)

// {

// if (n>0 && args[i][j] == k)

// {

// args[i] = args[i].Insert(j+1, '.');

// }

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m17(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 18 удалить все слова, в которых четное количество символов

//StringBuilder sb1 = new StringBuilder("Пушкин Александр Сергеевич");

//StringBuilder sb2 = new StringBuilder("Василевский Евгений Валерьевич");

//StringBuilder sb3 = new StringBuilder("Абражевич Александр Владимирович");

//StringBuilder sb4 = new StringBuilder("Розов Никита Сергеевич");

//StringBuilder res = new StringBuilder();

//StringBuilder m18(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// int n = 0;

// for(int j = 0; j < args[i].Length; j++)

// {

// for(char k = 'А'; k <= 'я'; k++)

// {

// if(k == args[i][j])

// {

// n++;

// }

// }

// if (args[i][j]==' ')

// {

// if (n % 2 == 0)

// {

// args[i] = args[i].Remove(j-n, n);

// }

// else { n = 0; }

// }

// }

// if (n % 2 == 0)

// {

// args[i] = args[i].Remove(args[i].Length-n, n);

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m18(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 19 заменить все английские заглавные буквы на \*

//StringBuilder res = new StringBuilder();

//StringBuilder m19(out StringBuilder result, params StringBuilder[] args)

//{

// result = new StringBuilder();

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

// {

// for (int j = 0; j < args[i].Length; j++)

// {

// for (char k ='A' ; k < 'Z'; k++)

// {

// if (k == args[i][j]) { args[i][j] = '\*'; }

// }

// }

// }

// result = new StringBuilder(String.Join<StringBuilder>(';', args));

// return result;

//}

//m19(out res, sb1, sb2, sb3, sb4);

//Console.WriteLine(res);

//Вариант 20 заменить все русские строчные буквы на \*

StringBuilder res = new StringBuilder();

StringBuilder m20(out StringBuilder result, params StringBuilder[] args)

{

result = new StringBuilder();

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

{

for (int j = 0; j < args[i].Length; j++)

{

for (char k = 'а'; k < 'я'; k++)

{

if (k == args[i][j]) { args[i][j] = '\*'; }

}

}

}

result = new StringBuilder(String.Join<StringBuilder>(';', args));

return result;

}

m20(out res, sb1, sb2, sb3, sb4);

Console.WriteLine(res);

Задание 3

using System.Text.RegularExpressions;

//Console.WriteLine("Удалить все русские буквы");

//string str = "Hello, 75634 Привет world проаврпов";

//Console.WriteLine(str);

//Console.WriteLine(m20(str));

//string m20(string s)

//{

// Regex regex = new Regex("[а-яА-ЯёЁ]");

// return regex.Replace(s, "");

//}

//Вариант 1 Найти только те слова сообщения,

//которые содержат не более чем n букв.

//string str = "Ночь, улица, dghddgfg фонарь, аптека.Бессмысленный и тусклый свет.";

//string[] m1(string s, int n)

//{

// string regex =$@"\b\w{{1,{n}}}\b";

// MatchCollection m2=Regex.Matches(str,regex);

// var words= new string[m2.Count];

// for(int i = 0; i < m2.Count; i++)

// {

// words[i] = m2[i].Value;

// }

// return words;

//}

//int n=Convert.ToInt16(Console.ReadLine());

//string[] m2=m1(str,n);

//foreach (var m in m2)

//{

// Console.WriteLine(m);

//}

//Вариант 2 Найти только те слова сообщения, которые начинаются

//с прописной буквы.

//string str = "Ночь, улица, dghddgfg фонарь, Аптека. Бессмысленный и Тусклый свет.";

//string[] m2(string s)

//{

// string regex = $@"\b[А-Я]\w+\b";

// MatchCollection m2 = Regex.Matches(str, regex);

// var words = new string[m2.Count];

// for (int i = 0; i < m2.Count; i++)

// {

// words[i] = m2[i].Value;

// }

// return words;

//}

//string[] m3 = m2(str);

//foreach (var m in m3)

//{

// Console.WriteLine(m);

//}

//Вариант 3 Найти только те слова сообщения,

//которые содержат хотя бы одну цифру.

//string str = "Но5чь, ули3ца, dghddgfg фонарь, Апт3ека. Бессмы3333сленный и Тусклый свет.";

//string[] m3(string s)

//{

// string regex = $@"\b\w+\d+\w+\b";

// MatchCollection m2 = Regex.Matches(str, regex);

// var words = new string[m2.Count];

// for (int i = 0; i < m2.Count; i++)

// {

// words[i] = m2[i].Value;

// }

// return words;

//}

//string[] m4 = m3(str);

//foreach (var m in m4)

//{

// Console.WriteLine(m);

//}

//Вариант 4 Удалить из сообщения все слова,

//которые заканчиваются на заданный символ.

//string str = "Ночь, улица, dghddgfg фонарь, Аптека. Бессмысленный и Тусклый свет.";

//string m4(string s, char c)

//{

// string reg =$@"\b\w+{c}\b";

// return Regex.Replace(s,reg, "").Trim();

//}

//char c = Console.ReadKey().KeyChar;

//Console.WriteLine();

//Console.WriteLine(m4(str, c));

//Вариант 5 Удалить из сообщения все слова,

//содержащие данный символ (без учета регистра)

//string str = "Ночь, улица, dghddgfg фонарь, АптЬека. Бессмысленный и Тусклый свет.";

//string m5(string s, char c)

//{

// string reg = $@"\b\w\*{c}\w\*\b";

// return Regex.Replace(s, reg, "", RegexOptions.IgnoreCase).Trim();

//}

//char c = Console.ReadKey().KeyChar;

//Console.WriteLine();

//Console.WriteLine(m5(str, c));

//Вариант 6 Удалить из сообщения все

//однобуквенные слова (вместе с лишними пробелами).

//string str = "Ночь, улица, у dghddgfg фонарь, АптЬека. Бессмысленный и Тусклый свет.";

//string m6(string s)

//{

// string reg = @"\b\w{1}\b";

// string cl=Regex.Replace(s, reg, "").Trim();

// cl = Regex.Replace(cl, @"\s+", " ");

// return cl;

//}

//Console.WriteLine();

//Console.WriteLine(m6(str));

//Вариант 7 Удалить из сообщения все английские слова (без учета регистра)

//string str = "Ночь, улица, dghddgfg фонарь,run, АптЬека. Бессмысленный и Тусклый свет.";

//string m7(string s)

//{

// string reg = $@"[a-zA-Z]";

// return Regex.Replace(s, reg, "", RegexOptions.IgnoreCase).Trim();

//}

//Console.WriteLine();

//Console.WriteLine(m7(str));

//Вариант 8 Подсчитать сколько раз заданное слово встречается в сообщении.

//string str = "Ночь, улица, dghddgfg фонарь,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//int m8(string s, string s1)

//{

// MatchCollection m= Regex.Matches(s, s1);

// return m.Count;

//}

//string str1 = "улица";

//Console.WriteLine();

//Console.WriteLine(m8(str,str1));

//Вариант 9 Подсчитать сколько слов,

//состоящих только из прописных букв, содержится в сообщении.

//string str = "НОЧЬ, УЛИЦА, dghddgfg ФОНАРЬ,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//int m9(string s)

//{

// string reg = $@"\b[A-ZА-Я]+\b";

// MatchCollection m = Regex.Matches(s, reg);

// return m.Count;

//}

//Console.WriteLine();

//Console.WriteLine(m9(str));

//Вариант 10 Найти все слова, содержащие цифры и английские буквы.

//string str = "НОЧЬ, УЛИЦА, dghd6746dgfg ФОНgdsf9АРЬ,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m10(string s)

//{

// string reg = $@"\b\w+\d+\w+\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas=new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas=m10(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 12 Найти все e-mail

//string str = "НОЧЬ, УЛИЦА, dghd6746dgfg@mail.ru ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m10(string s)

//{

// string reg = @"\b\w+\d\*\@\w+\.[A-Za-z]{2,3}\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m10(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 13 Найти все телефонные номера в формате ххх-хх-хх

//string str = "НОЧЬ,311-32-24 УЛИЦА, dghd6746dgfg@mail.ru 111-22-33 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m13(string s)

//{

// string reg = @"\b\d{3}-\d{2}-\d{2}\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m13(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 14 Найти все ip-адреса

//string str = "НОЧЬ,113.3.0.3 УЛИЦА, dghd6746dgfg@mail.ru 128.4.0.0 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m14(string s)

//{

// string reg = @"\b(\d{1,3}\.){3}\d{1,3}\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m14(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 15 Найти все слова, начинающиеся

//с заглавной буквы (остальные буквы строчные)

//string str = "Ночь,113.3.0.3 Улица, dghd6746dgfg@mail.ru 128.4.0.0 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m15(string s)

//{

// string reg = $@"\b[A-ZА-Я][a-zа-я]+\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m15(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 16 Подсчитать количество русских гласных

//string str = "Ночь, улица, dghddgfg фонарь,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//int m16(string s)

//{

// string reg = @"[УЕЫАОЭЯИЮуеыаоэяию]";

// MatchCollection m = Regex.Matches(s, reg);

// return m.Count;

//}

//Console.WriteLine();

//Console.WriteLine(m16(str));

//Вариант 17 Найти все даты в формате хх/хх/хххх

//string str = "Ночь,11/30/3123 Улица, dghd6746dgfg@mail.ru 12/04/1000 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m15(string s)

//{

// string reg = @"\b\d{2}\/\d{2}\/\d{4}\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m15(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 18 Найти все время в формате хх:хх:хх

//string str = "Ночь,11:30:31 Улица, dghd6746dgfg@mail.ru 12/04/1000 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string[] m15(string s)

//{

// string reg = @"\b\d{2}\:\d{2}\:\d{2}\b";

// MatchCollection m = Regex.Matches(s, reg);

// string[] mas = new string[m.Count];

// for (int i = 0; i < m.Count; i++)

// {

// mas[i] = m[i].ToString();

// }

// return mas;

//}

//string[] mas = m15(str);

//foreach (string s in mas)

//{

// Console.WriteLine(s);

//}

//Вариант 19 Удалить повторяющиеся пробелы.

//string str = "Ночь,11:30:31 Улица, dghd6746dgfg@mail.ru 12/04/1000 ФОНgdsf9АРЬ@mail.ru,run, улица АптЬека. Бессмысленный улица и Тусклый свет.";

//string m19(string s)

//{

// string reg = @"\s+";

// string cl = Regex.Replace(s, reg, " ");

// return cl;

//}

//Console.WriteLine( m19(str));

Задание 4

//Console.WriteLine("Входные данные - нет. Выходные данные - кол-во минут до конца месяца");

//Console.WriteLine(m20());

//int m20()

//{

// DateTime now = DateTime.Now;

// DateTime next = now.AddMonths(1);

// DateTime result = new DateTime(next.Year, next.Month, 1, 0, 0, 0);

// TimeSpan timespan = result - now;

// return (int)timespan.TotalMinutes;

//}

//Вариант 1 Входные данные – количество лет (int).

//Выходные – день недели, на которые выпадет

//текущая дата через указанное количество лет (string).

//string m1(int n)

//{

// DateTime now = DateTime.Now;

// DateTime date1 = now.AddYears(n);

// return date1.DayOfWeek.ToString();

//}

//Console.WriteLine(m1(7));

//Вариант 2 Входные данные – дата рождения (string).

//Выходные данные – количество дней до текущей даты(int).

//int m2(string s)

//{

// DateTime now = DateTime.Now;

// DateTime s1 = DateTime.Parse(s);

// TimeSpan s3 = now - s1;

// return (int) s3.TotalMinutes/(24\*60);

//}

//string str= "1 декабря 1993";

//Console.WriteLine(m2(str));

//Вариант 3 Входные данные – время (string).

//Выходные данные – количество секунд до текущего момента

//(int, возможно отрицательное).

//int m3(string s)

//{

// DateTime now = DateTime.Now;

// DateTime s1 = DateTime.Parse(s);

// TimeSpan s3 = now - s1;

// return (int)s3.TotalSeconds;

//}

//string str = "16:30:23";

//Console.WriteLine(m3(str));

//Вариант 4 Входные данные – дата и время (string).

//Выходные данные – количество минут до ближайшего Нового года (int).

//int m4(string s)

//{

// DateTime date = DateTime.Parse(s);

// DateTime next = date.AddYears(1);

// DateTime nres = new DateTime(next.Year,1,1,0,0,0);

// DateTime lres = new DateTime(date.Year, 1, 1, 0, 0, 0);

// if (date.DayOfYear < 183)

// {

// TimeSpan t = date - lres;

// return (int)t.TotalMinutes;

// }

// else

// {

// TimeSpan g = nres - date;

// return (int)g.TotalMinutes;

// }

//}

//string str = "1 декабря 2024 18:30";

//Console.WriteLine(m4(str));

//Вариант 5 Входные данные – время начала занятий (string).

//Выходные данные – время окончания занятий

//(4 урока по 45 минут, 15 минут перерыв, TimeOnly).

//TimeOnly m5(string s)

//{

// TimeOnly t= TimeOnly.Parse(s);

// int y = 4;

// int p = 3;

// for(int i = 0; i < y; i++)

// {

// t=t.AddMinutes(45);

// }

// for (int i = 0; i < p; i++)

// {

// t = t.AddMinutes(15);

// }

// return t;

//}

//string str = "11:00";

//Console.WriteLine(m5(str));

//Вариант 6 Входные данные – дата начала проекта (string).

//Выходные данные – дата окончания проекта(dd: mm:yyyy).

//На проект предусмотрено 1000 календарных дней.

//string m6(string s)

//{

// DateTime date = DateTime.Parse(s);

// return date.AddDays(1000).ToString("dd: MM:yyyy");

//}

//string str = "31 декабря 2023";

//Console.WriteLine(m6(str));

//Вариант 7 Входные данные – дата (string).

//Выходные данные – количество дней до конца месяца (int).

//int m7(string s)

//{

// DateTime data = DateTime.Parse(s);

// DateTime next = data.AddMonths(1);

// DateTime nres = new DateTime(next.Year, next.Month, 1,0,0,0);

// TimeSpan t = nres - data;

// return (int)t.TotalDays;

//}

//string str = "31 декабря";

//Console.WriteLine(m7(str));

//Вариант 8 Входные данные – дата (string).

//Выходные данные – количество дней до конца недели (int).

//int m8(string s)

//{

// DateOnly date = DateOnly.Parse(s);

// if(date.DayOfWeek == DayOfWeek.Sunday)

// {

// return 1;

// }

// int c = 2;

// for(DayOfWeek dow = date.DayOfWeek; dow < DayOfWeek.Saturday; dow++)

// { c++; }

// return c;

//}

//string str = "7 декабря 2024";

//Console.WriteLine(m8(str));

//Вариант 9 Входные данные – время (string).

//Выходные данные – количество минут до полуночи (int).

//int m9(string s)

//{

// TimeOnly t=TimeOnly.Parse(s);

// TimeOnly pol = new TimeOnly(0,0,0);

// TimeSpan t1 = pol - t;

// return (int)t1.TotalMinutes;

//}

//string str = "19:30";

//Console.WriteLine(m9(str));

//Вариант 10 Входные данные – время (string).

//Выходные данные – количество секунд до полуночи (int).

//int m9(string s)

//{

// TimeOnly t = TimeOnly.Parse(s);

// TimeOnly pol = new TimeOnly(0, 0, 0);

// TimeSpan t1 = pol - t;

// return (int)t1.TotalSeconds;

//}

//string str = "19:30:23";

//Console.WriteLine(m9(str));

//Вариант 11 Входные данные – время выполнения задачи в тиках (int).

//Выходные данные – сколько раз может быть выполнена

//задача до конца текущего часа (int).

//int m11(long t)

//{

// DateTime now = DateTime.Now;

// DateTime next = now.AddHours(1);

// DateTime nres = new DateTime(now.Year, now.Month, now.Day, next.Hour, 0, 0,0);

// TimeSpan t1 = nres - now;

// long t2=(long)t1.TotalMinutes;

// long t3 = (long)t1.TotalSeconds;

// long t4 = (long)t1.TotalMilliseconds;

// return (int)(((t2 \* 60 \* 10000000) + (t3 \* 10000000) + t4) / t);

//}

//long k = 9232323;

//Console.WriteLine(m11(k));

//Вариант 12 Входные данные – дата рождения (string).

//Выходные данные – на какой день недели выпал бы ваш день рождения,

//если бы вы родились на 100 лет раньше (string).

//DayOfWeek m12(string s)

//{

// DateTime date = DateTime.Parse(s);

// DateTime dater = date.AddYears(-100);

// return dater.DayOfWeek;

//}

//string str = "1 декабря 1993";

//Console.WriteLine(m12(str));

//Вариант 14 Входные данные – дата (string).

//Выходные данные – какой день недели будет ровно 3 месяца спустя(string).

//string m14(string s)

//{

// DateTime date = DateTime.Parse(s);

// DateTime datem = date.AddMonths(3);

// return datem.DayOfWeek.ToString();

//}

//string str = "1 сентября 2024";

//Console.WriteLine(m14(str));

//Вариант 15 Входные данные – дата зачатия ребенка (string).

//Выходные данные – предполагаемая дата рождения

//(беременность длится 40 недель, string).

//string m14(string s)

//{

// DateTime date = DateTime.Parse(s);

// DateTime datem = date.AddMonths(10);

// return datem.ToString();

//}

//string str = "1 сентября 2024";

//Console.WriteLine(m14(str));

//Вариант 16 Входные данные – дата зачатия ребенка (string).

//Выходные данные – предполагаемый день недели рождения ребенка

//(беременность длится 40 недель, string).

//string m14(string s)

//{

// DateTime date = DateTime.Parse(s);

// DateTime datem = date.AddMonths(10);

// return datem.DayOfWeek.ToString();

//}

//string str = "1 сентября 2024";

//Console.WriteLine(m14(str));

//Вариант 17 Входные данные – дата и время (string).

//Выходные данные – сколько минут прошло до текущего момента.

//int m17(string s)

//{

// DateTime now = DateTime.Now;

// DateTime date = DateTime.Parse(s);

// TimeSpan t = now - date;

// return (int)t.TotalMinutes;

//}

//string str = "21 ноября 2024";

//Console.WriteLine(m17(str));

//Вариант 18 Входные данные – время окончания занятия (string).

//Выходные данные – количество секунд до этого момента.

//int m17(string s)

//{

// DateTime now = DateTime.Now;

// DateTime date = DateTime.Parse(s);

// TimeSpan t = date-now;

// return (int)t.TotalSeconds;

//}

//string str = "18:00";

//Console.WriteLine(m17(str));

//Вариант 19 Входные данные – нет.

//Выходные данные – количество часов до полуночи.

int m19()

{

DateTime now = DateTime.Now;

DateTime pol = now.AddDays(1);

DateTime next = new DateTime(now.Year, now.Month, pol.Day, 0, 0, 0);

TimeSpan t = next - now;

return (int)t.TotalHours;

}

Console.WriteLine(m19());