**二. 思考题**

**4.**

随机产生字符数字

**5.**

字符串是用于表示文本的字符的有序集合。

C#分为常规字符串文本和逐字字符串文本两类

**6.**

Concat()方法：整体是一个数组的拷贝，虽然在内存中是处理都是原子性操作，速度非常快，但是，最后的return语句创建一个新String对象，也就是每次concat操作都会创建一个新的String对象，这也是限制concat方法速度的原因。

Append()方法：整个append方法都在做字符数组的处理，加长，拷贝等，这些都是基本的数据处理，整个方法内并没有生成对象。只是最后toString返回一个对象而已。需要注意的是，append()方法返回的是一个StringBuilder(or StringBuffer)对象实例。

**三．编程题**

**2.**

using System;

namespace text2

{

class Program

{

static void Main(string[] args)

{

string s = "asdasd";

Console.Write("Before deletion: s=");

Console.WriteLine(s);

Console.WriteLine("\nInput the char that you want to delete in s.");

string a = Console.ReadLine().ToString();

Console.Write("\nAfter deletion: s=");

s = s.Replace(a, "");

Console.WriteLine(s);

Console.WriteLine("\nPress any key to quit.");

Console.ReadKey();

}

}

}

**3.**

using System;

using System.Collections.Generic;

namespace text3

{

class Program

{

static void Main(string[] args)

{

//用栈和队列

Console.WriteLine("请输入要进行判断的字符串");

string str = Console.ReadLine();

Stack<char> stack = new Stack<char>();

Queue<char> queue = new Queue<char>();

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

{

stack.Push(str[i]);

queue.Enqueue(str[i]);

}

bool isPlalindrome = true;//isPlalindrome初始化为真

while (stack.Count > 0)

{

if (stack.Pop() != queue.Dequeue())//只要发现有一个不等，就把isPlalindrome设置为假

{

isPlalindrome = false;

break;//发现有一个不等就退出循环

}

}

Console.WriteLine("字符串是回文串：" + isPlalindrome);

Console.ReadKey();

}

}

}

**13.**

using System;

namespace text13

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Input 10 integers...");

int[] a = new int[10];

for (int i = 0; i < 10; i++)

a[i] = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Before exchange :");

foreach (var i in a)

Console.Write(i + ",");

for (int i = 0; i < 5; i++)

{

int t = a[i];

a[i] = a[9 - i];

a[9 - i] = t;

}

Console.WriteLine("\nAfter exchange :");

foreach (var i in a)

Console.Write(i + ",");

Console.WriteLine("\nPress any key to quit.");

Console.ReadKey();

}

}

}