先来个小教训：

textBox2.Text += A[i].ToString();//注意这里是 “+” 号

就相当于 字符串赋值，如果没有加号，每一循环都会 重新赋值，以前的值就会丢失

有了加号，就把输出的内容 全部保存起来，然后全部输出到窗体。

private void button1\_Click(object sender, EventArgs e)

{

int N,R=0;

int[] A = new int[200];

N = Convert.ToInt32(textBox1.Text);

for (int i = 1; i <= N; ++i)

{

if (i%2!=0)

{

A[R++] = i;

}

}

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

{

textBox2.Text += A[i].ToString();//注意这里是 “+” 号

textBox2.Text += " ";

}

}

操作：Console.WriteLine ( n[ I , j ]+” ” ); 实际作用嘛，自己理解咯

收获：

1. Object ：所有类的祖宗
2. 创建数组：int[ ] a = new int [3]; 需要知道的是，int[ ] 是一种数据类型，专门用于数组。

int[ , ] n = new int [3,2] { {1,2},{3,4},{5,6}}; a[ I , j ] == a[ I ][ j ];

1. 可能从没注意到，VS中也 注释选中行的功能
2. 还记得 C语言的循环吗，do…while（）; 要记得这里循环后面有分号哦
3. foreach ：对数组操作的另一种形式

Eg：foreach (int k,In a ) { xxxxx }

其含义是：利用K在数组a中寻找有实际含义的整型变量，多维数组操作更方便

Ok，现在把压轴的拿上来：

引例：在很多可能下，我们有可能某些输入让程序崩溃 比如：输入字符型，转换为整型

这时候，你可能以为，记得怎么大体的输入方式就行了呗..

No！

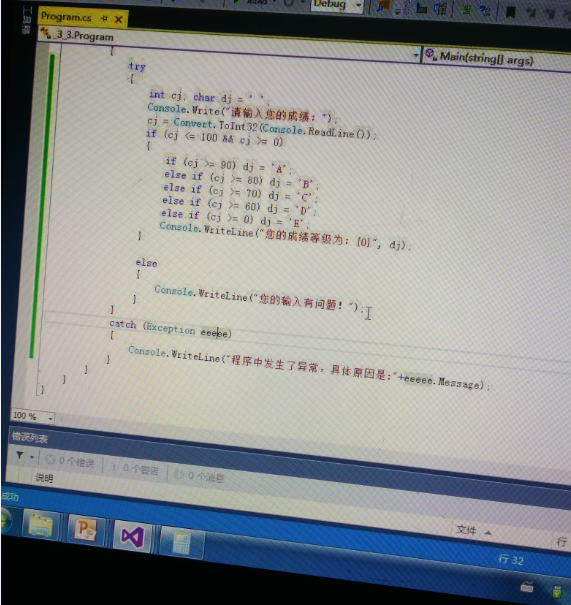
因为输入的对象是顾客，所以输入情况不可能一样，这个时候你就要做些骚操作了

try { x大体操作x }

catch { Console.WriteLine( 您的输入有误zzz..);}

finaly{ Console.WriteLine( 您的输入有误zzz..);}

讲下含义，try 的花括号里面便是，可能使程序崩溃的操作，比如下面：



很有可能出现 ，输入字符转换整型的情况。

try，“试一试“，把可能犯错的操作 括起来！

catch：“抓住，扑捉”，抓住程序崩溃之处，输出骚话！

finaly：“最后的”，最后再来一补刀，与 catch 操作类似！

记住：必须要有try，然后必须还要有 catch，finaly，其中一个。当然都有更好

2017年3月15日19:42:51，时间还早，加油！