

1 把一个长度为4的数组 添加一个新数据

注意：数组的长度不能修改 增加长度必须重新开辟新空间new

```
1      int[] x = { 100, 200, 90, 29 };
2      //y保存x数组起始数据
3      int[] y = x;
4
5      x = new int[y.Length + 1];
6
7      Console.WriteLine(x[0]);
8
9      for (int i = 0; i < y.Length; i++)
10     {
11         x[i] = y[i];
12     }
13
14     x[x.Length - 1] = 500;
15     Console.WriteLine(x[0]);
16     Console.Write(x[4]);
```

2 把一个长度为4的数组 插入一个新数据

array 10 20 30 40 50

array2 10 20 30 40 50

array 0 0 0 0 0 0

array 10 20 30 0 0 0

array 10 20 30 data 50 0

```
1      int[] array = { 10, 20, 30, 40,50 };
2
3      Console.WriteLine("请输入需要插入数据的位置");
4      int index = int.Parse(Console.ReadLine());
5
6      Console.WriteLine("请输入需要插入的数据");
7      int data = int.Parse(Console.ReadLine());
8
```

```

9
10         int x = 0;
11
12         int[] array2 = array;
13         array = new int[array.Length + 1];
14
15         for (int i = 0; i < index; i++)
16         {
17             array[i] = array2[x++];
18         }
19         array[index] = data;
20
21         for (int i = index + 1; i < array.Length; i++)
22         {
23             array[i] = array2[x++];
24         }
25
26         for (int i = 0; i < array.Length; i++)
27         {
28             Console.WriteLine(array[i]);
29         }
30

```

3 把1个int类型的数组从小到大排序

冒泡排序算法

每次外层循环 确定一个位置的数

即 第一次大循环确定最小值 放在数组的第一个元素

每次确定的数 不参与下次大循环的比较

```

1         int[] x = { 7, 17, 20, 22, 90 };
2
3         for (int i = 0; i < x.Length; i++)
4         {
5             for (int j = i; j < x.Length; j++)
6             {

```

```

7         if (x[j] < x[i])
8         {
9             int a = x[i];
10            x[i] = x[j];
11            x[j] = a;
12        }
13    }
14 }
15
16 for (int i = 0; i < x.Length; i++)
17     Console.WriteLine(x[i]);

```

多维数组 [,]

一般使用的是二维数组

```

1         int[, ] x = new int[3, 4];
2
3         for (int i = 0; i < x.GetLength(0); i++)
4         {
5             for (int j = 0; j < x.GetLength(1); j++)
6             {
7                 x[i, j] = i * j;
8             }
9         }
10
11        for (int i = 0; i < x.GetLength(0); i++)
12        {
13            for (int j = 0; j < x.GetLength(1); j++)
14            {
15                Console.Write(x[i, j] + "\t");
16            }
17            Console.WriteLine();
18        }

```

边声明边赋值写法：

```

1      int[,] x = new int[,]{{1,2,3,4},{2,2,3,3},
{900,110,123,1}};
2      int[,] y = { { 1, 2, 3,}, {2, 2, 2, 3 }, { 900, 110, 123,
1 } } };

```

注意：在使用多维数组时 二维数组的每一行的元素个数必须是一样的

例：

用一个10*10的二维数组 表示一张地图

在地图的最外围显示"★" 内部显示"☆"

```

1  string[,] strs = new string[10, 10];
2
3      for (int i = 0; i < strs.GetLength(0); i++)
4      {
5          for (int j = 0; j < strs.GetLength(1); j++)
6          {
7              if (i == 0 || i == strs.GetLength(0)-1
8                  || j == 0 || j == strs.GetLength(1)-1)
9              {
10                 strs[i, j] = "★";
11             }else
12                 strs[i, j] = "☆";
13             }
14         }
15
16     for (int i = 0; i < strs.GetLength(0); i++)
17     {
18         for (int j = 0; j < strs.GetLength(1); j++)
19         {
20             Console.Write(strs[i, j]);
21         }
22         Console.WriteLine();
23     }

```


输出W型

```
1      string[,] strs = new string[5, 60];
2
3      for (int i = 0; i < strs.GetLength(0); i++)
4      {
5          for (int j = 0; j < strs.GetLength(1); j++)
6              strs[i, j] = "☆";
7      }
8
9      int x = 0;
10     bool add = true;
11
12     for (int j = 0; j < strs.GetLength(1); j++)
13     {
14         strs[x, j] = "★";
15         if (add)
16         {
17             x++;
18             if (x == strs.GetLength(0))
19             {
20                 x--;
21                 add = false;
22             }
23         }
24         else
25         {
26             x--;
27             if (x < 0)
28             {
29                 x++;
30                 add = true;
31             }
32         }
33     }
34
35     for (int i = 0; i < strs.GetLength(0); i++)
36     {
37         for (int j = 0; j < strs.GetLength(1); j++)
38             Console.Write(strs[i, j]);
39         Console.WriteLine();
```

运行效果：



```

C:\Windows\system32\cmd.exe
★☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★
☆☆★☆☆☆☆☆☆☆☆★☆☆★☆☆☆☆☆☆☆☆★☆☆★☆☆☆☆☆☆☆☆★☆☆
☆☆☆☆★☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★
☆☆☆☆★☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆★☆☆☆☆☆☆☆☆★☆☆☆☆★☆☆☆☆
☆☆☆☆★☆☆★☆☆☆☆☆☆☆☆☆☆☆☆☆☆★☆☆☆☆☆☆☆☆★☆☆★☆☆☆☆
请按任意键继续...
  
```

练习：

使用二维数组 设定一张地图

使用wasd控制地图上的人物上下左右移动

```

1      string[,] map = new string[10, 10];
2
3      for (int i = 0; i < map.GetLength(0); i++)
4      {
5          for (int j = 0; j < map.GetLength(1); j++)
6          {
7              if (i == 0 || i == map.GetLength(0) - 1
8                  || j == 0 || j == map.GetLength(1) - 1)
9              {
10                 map[i, j] = "☆☆";
11             }
12             else
13             {
14                 map[i, j] = "  ";
15             }
16         }
17     }
18
19     int hero_x=1;
  
```

```
20     int hero_y=1;
21
22     map[hero_x, hero_y] = "★";
23
24     while (true)
25     {
26         Console.Clear();
27
28         //打印地图
29         for (int i = 0; i < map.GetLength(0); i++)
30         {
31             for (int j = 0; j < map.GetLength(1); j++)
32                 Console.Write(map[i, j]);
33             Console.WriteLine();
34         }
35
36         Console.WriteLine("请输入wasd控制人物上下左右移动");
37
38         char cmd = (char)Console.Read();
39
40         switch (cmd)
41         {
42             case 'w':
43                 if (hero_x > 1)
44                 {
45                     map[hero_x, hero_y] = " ";
46                     map[--hero_x, hero_y] = "★";
47                 }
48                 break;
49             case 'a':
50                 if (hero_y > 1)
51                 {
52                     map[hero_x, hero_y] = " ";
53                     map[hero_x, --hero_y] = "★";
54                 }
55                 break;
56             case 's':
57                 if (hero_x < map.GetLength(0) - 2)
58                 {
59                     map[hero_x, hero_y] = " ";
60                     map[++hero_x, hero_y] = "★";
61                 }
62             }
```



```

62         break;
63     case 'd':
64         if (hero_y < map.GetLength(1) - 2)
65         {
66             map[hero_x, hero_y] = " ";
67             map[hero_x, ++hero_y] = "★";
68         }
69         break;
70     }
71 }
72

```

交叉数组 [][]

一个一维数组中的每个元素都是第一个数组

交叉数组每一行的元素个数可以不一样 这是和二维数组的不同点

定义 先定义行 再定义列

```

1      int[][] x=new int[2][];
2      x[0] = new int[] { 1, 2, 3 };
3      x[1] = new int[] { 11, 22, 33, 44, 55, 66, 77 };
4
5      for (int i = 0; i < x.Length; i++)
6      {
7          for (int j = 0; j < x[i].Length; j++)
8          {
9              Console.Write(x[i][j] + "\t");
10             }
11             Console.WriteLine();
12     }

```