第11讲 字符串处理

教材10.1~10.5节

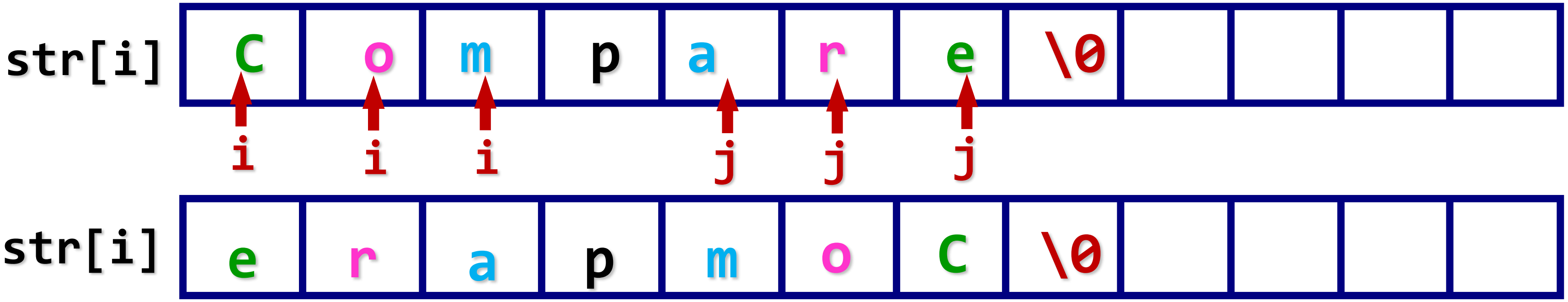
MOOC第10周

哈尔滨工业大学

苏小红
sxh@hit.edu.cn

用字符数组实现字符串逆序

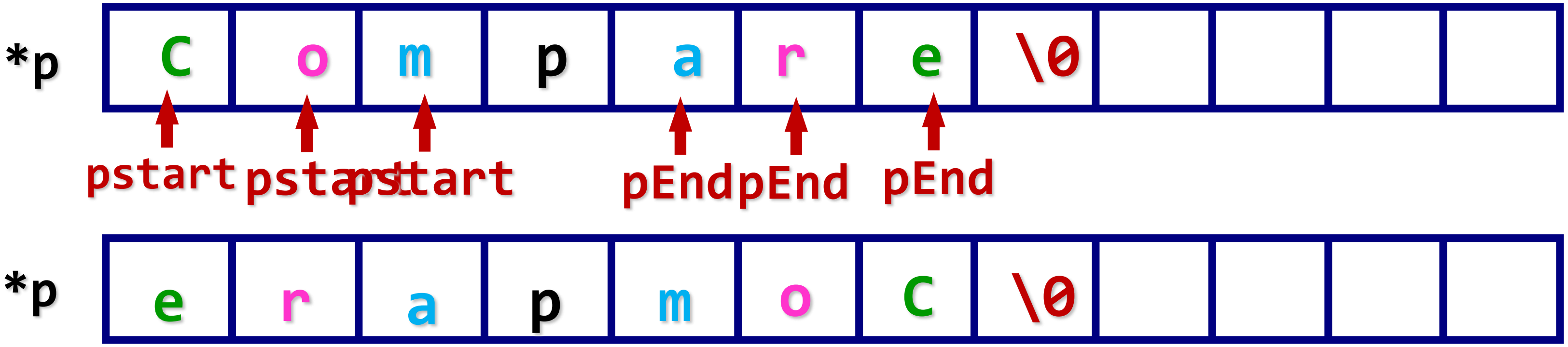
```
void Inverse(char str[])
{
   int   len, i, j;
   char  temp;
   len = strlen(str);
   for (i=0, j=len-1; i<j; i++, j--)
   {
        temp = str[i];
       str[i] = str[j];
       str[j] = temp;
   }
}
```



str[i]	C	o	m	p	a	r	e	\0				
	i	i	i		j	j	j					
str[i]	e	r	a	p	m	o	C	\0				

用字符指针实现字符串逆序

```
void Inverse(char *p)
{
   int  len;
   char temp, char *pStart, *pEnd;
   len = strlen(p);
   for(pStart=p,pEnd=p+len-1; pStart<pEnd; pStart++,pEnd--)
    {
       temp = *pStart;
       *pStart = *pEnd;
       *pEnd = temp;
   }
}
```



*p | C | o | m | p | a | r | e | \0 | | | |

pstart pstart pstart pEnd pEnd pEnd

*p | e | r | a | p | m | o | C | \0 | | | |

用字符数组实现字符串逆序

```
void Inverse(char str[])
{
    int   len, i;
    char  temp;
    len = strlen(str);
    for (i=0; i<len/2; i++)
    {
         temp = str[i];
        str[i] = str[len-1-i];
        str[len-1-i] = temp;
    }
}
```

str[i]	C	o	m	p	a	r	e	\0				

str[i]	e	r	a	p	m	o	C	\0				

讨论

- 在英语中，与回文诗对应的是回文词，无论是从前往后拼读，还是从后往前拼读，它们的拼法和词义都不变。
  - 例如：dad（爸爸），mum（妈妈），noon（中午），eye（眼睛）等
- 从键盘任意输入一个字符串，编程判断这个字符串是否是“回文”。如果是回文，则输出"Yes!"，否则输出"No!"。
- 想一想，回文字符串判断程序和字符串逆序程序有什么联系？

```
void Inverse(char str[])    //字符串逆序
{
     int    i, j;
     char   temp;
    for (i=0, j=strlen(str)-1; i<j; i++, j--)
     {
            temp = str[i];
            str[i] = str[j];
            str[j] = temp;
     }
}
```

1	2	3	4	5	6

6	5	4	3	2	1

```
int Plalindrome(char str[]) //回文字符串
{
        int i, j;
        for (i=0, j=strlen(str)-1; i<j; i++, j--)
        {
                if (str[i] != str[j]) //互换改成比较
                {
                        return 0;
                }
        }
        return 1;
}
```

1	2	3	3	2	1

1	2	3	4	5	1

统计单词个数

```
//统计str中单词的个数
int CountWords(char str[])
{
    int  i, num = 0;
    num = (str[0]!=' ') ? 1 : 0; //若无此语句，则配合新的返回值语句
    for (i=1; str[i]!='\0'; i++)
    {
        if (str[i]!=' ' && str[i-1]==' ')
        {
            num++;
        }
    }
    return num;  //return (str[0]!=' ') ? num+1 : num;
}
```

H	e	l	l	o		C	h	i	n	a	\0

	O	u	r			C	h	i	n	a	\0

统计单词个数

```
//统计str中单词的个数，需要#include <ctype.h>
int CountWords(char str[])
{
    int  i, num = 0;
    num = (!isspace(str[0]))? 1 : 0; //若无此语句则配合新的返回值语句
    for (i=1; str[i]!='\0'; i++)
    {
        if (!isspace(str[i]) && isspace(str[i-1]))
        {
            num++;
        }
    }
    return num;  //return (!isspace(str[0]) ? num+1 : num;
}
```

H	e	l	l	o		C	h	i	n	a	\0

	O	u	r			C	h	i	n	a	\0

字符处理函数

判断输入的字符串中是否包含数字、空格、英文字母、小写字母、大写字母

```
isdigit()
isspace()
isalpha()
islower()
isupper()
#include <ctype.h>
```

注意：返回值只有0和非0（不一定是1）

```
//num1[j] = s1[i] - '0'; //数字字符转换为纯数字
num1[j] = atoi(s1[i]);
#include <stdlib.h>
```

字符串的大小写转换

```
for (i=0; str[i]!='\0'; i++)
{
    if (str[i]>='A' && str[i]<='Z')
    {
        str[i] = str[i] + ('a'-'A');
    }
}
```

```
for (i=0; str[i]!='\0'; i++)
{
    if (str[i]>='a' && str[i]<='z')
    {
        str[i] = str[i] – ('a'-'A');
    }
}
```

```
for (i=0; str[i]!='\0'; i++)
{
    if (isupper(str[i]))
    {
        str[i] = tolower(str[i]);
    }
}
```

```
for (i=0; str[i]!='\0'; i++)
{
    if (islower(str[i]))
    {
        str[i] = touuper(str[i]);
    }
}
```

统计最长的单词的长度

```
//统计str中最长单词的长度，不考虑文本中存在非英文字符
int LongWordLenth(char str[])
{
    int  i, num = 0, max = 0;
    for (i=0; str[i]!='\0'; i++)
    {
        if (str[i]!=' ')  num++;
        else              num = 0;
        if (num > max)    max = num;
    }
    return max;
}
```

H	e		i	s		a		b	o	y	\0

课后作业

- 求一串文本中最后一个单词的长度
  - 1）不考虑文本中存在非英文字符
  - Hello World
  - 5
  - 2）考虑句末存在非英文字符
  - Hello World.
  - 5

课后作业

- 求一串文本中最后一个单词
  - 1）不考虑文本中存在非英文字符
  - Hello World
  - word
  - 2）考虑句末存在非英文字符
  - Hello World.
  - world

巧用字符数组求解数值问题

- 计算数字根。所谓数字根，是指把一个正整数的各位数字相加，若此和为一位数，则此和即为其根，否则把它的各位数字继续相加，直到为一位数为止。
  - 39，3+9=12，12不是一位数，1+2=3是一位数，其数字根是3。

```c
#include <stdio.h>
#include <string.h>
#define N 100
int DigitSum(int num);
int main()
{
    int n;
    printf("Input n:");
    scanf("%d", &n);
    do{
        n = DigitSum(n);
    }while (n/10 != 0);
    printf("%d\n", n);
    return 0;
}
```

```c
//计算num的各位数字之和并返回
int DigitSum(int num)
{
    int i = 0, n = 0, sum;
    int a[N];
    while (num != 0)
    {
        a[i] = num % 10;
        num = num / 10;
        i++;
    }
    n = i;
    for (sum=0,i=0; i<n; i++)
    {
        sum = sum + a[i];
    }
    return sum;
}
```

巧用字符数组求解数值问题

- 计算数字根。所谓数字根，是指把一个正整数的各位数字相加，若此和为一位数，则此和即为其根，否则把它的各位数字继续相加，直到为一位数为止。
  - 39，3+9=12，12不是一位数，1+2=3是一位数，其数字根是3。

```
//计算num的各位数字之和并返回
int DigitSum(int num)
{
    int i = 0, n = 0, sum;
    int a[N];
    while (num != 0)
    {
        a[i] = num % 10;
        num = num / 10;
        i++;
    }
    n = i;
    for (sum=0,i=0; i<n; i++)
    {
        sum = sum + a[i];
    }
    return sum;
}
```

```
//计算num的各位数字之和并返回
int DigitSum(int num)
{
    int i, sum;
    char a[N];
    //分离各位数字到字符数组a中
    sprintf(a, "%d", num);

    for (sum=0,i=0; a[i]!='\0'; i++)
    {
        sum = sum + (a[i] - '0');
    }
    return sum;
}
```

巧用字符数组求解数值问题

- 将1到9这九个数字分成三个3位数，要求第一个3位数，正好是第二个3位数的1/2，是第三个3位数的1/3。问应当怎样分。
- 程序的运行结果：

```
192,384,576

219,438,657

273,546,819

327,654,981
```

```c
#include <stdio.h>
#include <string.h>
int Separate(int m);
int main()
{
  int m;
  for (m=123; m<333; m++)
  {
    if (Separate(m))
    {
       printf("%d,%d,%d\n", m, m*2, m*3);
    }
  }
  return 0;
}
```

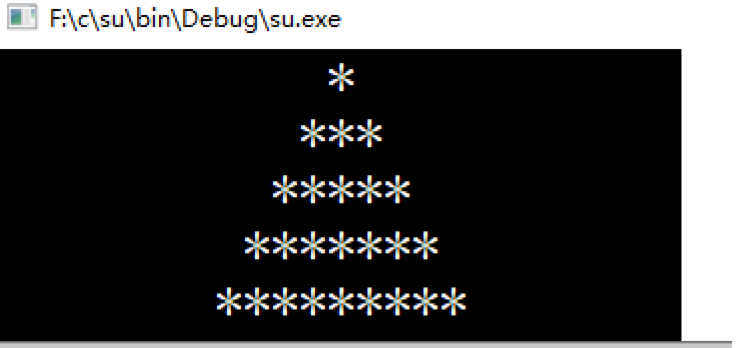
巧用字符数组求解数值问题

```
int Separate(int m)
{
    int i, j, a[9];
    a[0] = m / 100;
    a[1] = (m % 100) / 10;
    a[2] = m % 10;
    a[3] = (m * 2) / 100;
    a[4] = ((m * 2) % 100) / 10;
    a[5] = (m * 2) % 10;
    a[6] = (m * 3) / 100;
    a[7] = ((m * 3) % 100) / 10;
    a[8] = (m * 3) % 10;
    for (i=0; i<9; i++)
    {
        for (j=0; j<i; j++)
        {
            if ((a[i]==a[j])
                || a[i]==0 || a[j]==0)
            {
                return 0;
            }
        }
    }
    return 1;
}
```

```
int Separate(int m)
{
    int i, j;
    char a[9];
    sprintf(a, "%d", m);
    sprintf(a+3, "%d", 2*m);
    sprintf(a+6, "%d", 3*m);

    for (i=0; i<9; i++)
    {
        for (j=0; j<i; j++)
        {
          if ((a[i]==a[j])
              || a[i]=='0' || a[j]=='0')
          {
              return 0;
          }
        }
    }
    return 1;
}
```

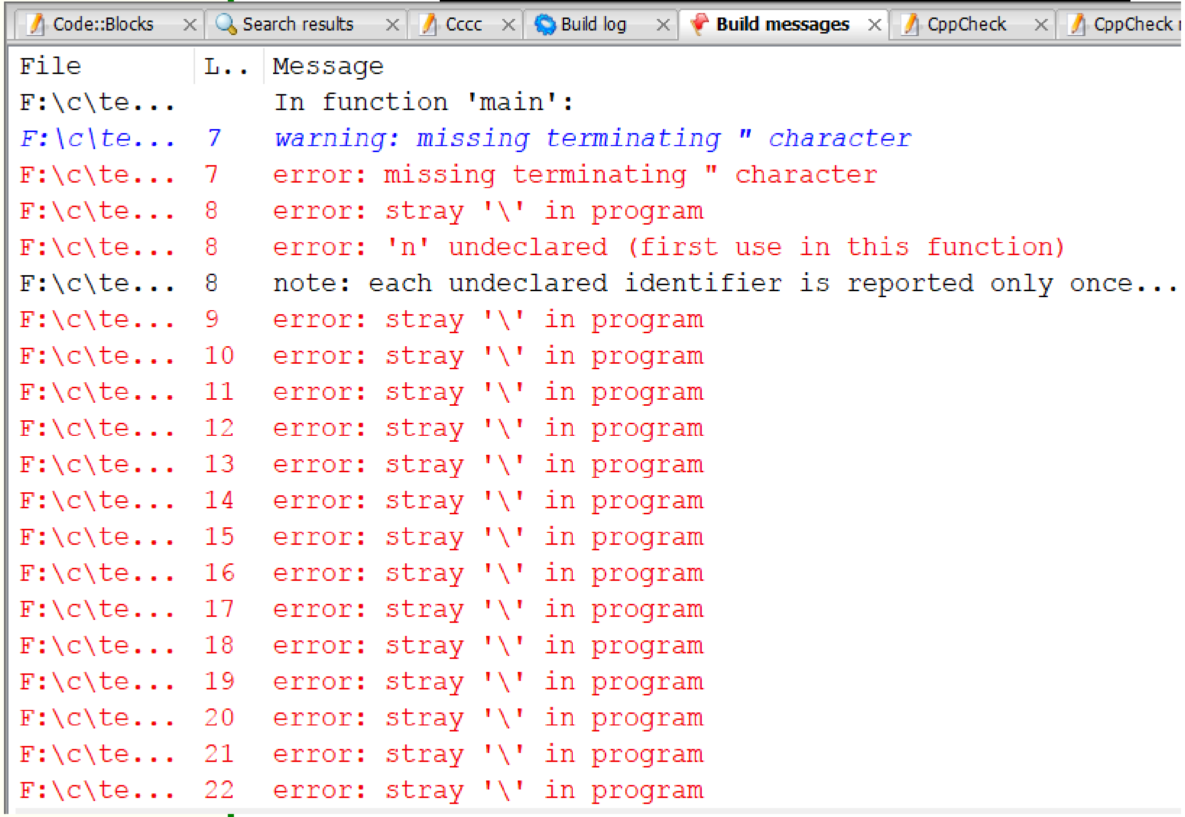
如何存储一个超长的字符串？

```c
#include <stdio.h>
#define N 1000
int main()
{
    char str[N] =
    {
        "            *
                    ***
                   *****
                  *******
                 *********
                   *****
                  *******
                 *********
                ***********
               *************
                 *********
                ***********
               *************
              ***************
             *****************
                    ***
                    ***
                    ***\n"
    };

    printf("%s", str);

    return 0;
}
```



F:\c\su\bin\Debug\su.exe

```
    *
   ***
  *****
 *******
*********
```



```
File         L..  Message
F:\c\te...        In function 'main':
F:\c\te...   7    warning: missing terminating " character
F:\c\te...   7    error: missing terminating " character
F:\c\te...   8    error: stray '\' in program
F:\c\te...   8    error: 'n' undeclared (first use in this function)
F:\c\te...   8    note: each undeclared identifier is reported only once...
F:\c\te...   9    error: stray '\' in program
F:\c\te...   10   error: stray '\' in program
F:\c\te...   11   error: stray '\' in program
F:\c\te...   12   error: stray '\' in program
F:\c\te...   13   error: stray '\' in program
F:\c\te...   14   error: stray '\' in program
F:\c\te...   15   error: stray '\' in program
F:\c\te...   16   error: stray '\' in program
F:\c\te...   17   error: stray '\' in program
F:\c\te...   18   error: stray '\' in program
F:\c\te...   19   error: stray '\' in program
F:\c\te...   20   error: stray '\' in program
F:\c\te...   21   error: stray '\' in program
F:\c\te...   22   error: stray '\' in program
```

```c
#include <stdio.h>
#define N 500

int main()
{
    char str[N] =
    {
        "            *\n"
        "           ***\n"
        "          *****\n"
        "         *******\n"
        "        *********\n"
        "          *****\n"
        "         *******\n"
        "        *********\n"
        "       ***********\n"
        "      *************\n"
        "        *********\n"
        "       ***********\n"
        "      *************\n"
        "     ***************\n"
        "    *****************\n"
        "           ***\n"
        "           ***\n"
        "           ***\n"
    };

    printf("%s", str);

    return 0;
}
```

```c
#include <stdio.h>
#define N 500

int main()
{
    char str[N] =
    {
        "            *\n"
        "           ***\n"
        "          *****\n"
        "         *******\n"
        "        *********\n"
        "          *****\n"
        "         *******\n"
        "        *********\n"
        "       ***********\n"
        "      *************\n"
        "        *********\n"
        "       ***********\n"
        "      *************\n"
        "     ***************\n"
        "    *****************\n"
        "           ***\n"
        "           ***\n"
        "           ***\n"
    };
    int i;
    for(i = 0; str[i]!='\0'; i++)
    {
        printf("%c", str[i]);
    }
    return 0;
}
```

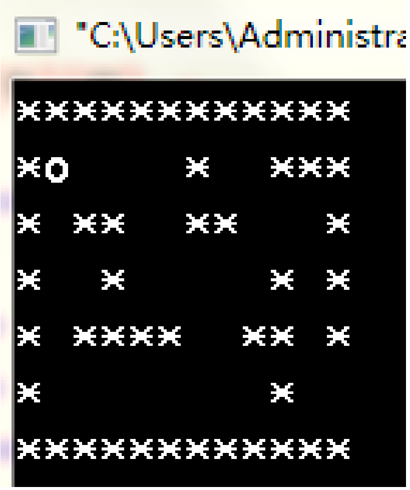
二维数组相当于
创建了数组的数组

```c
#include <stdio.h>
#define N 18
#define M 30
int main()
{
    char str[N][M] =
    {
str[0]  "            *\n",
str[1]  "           ***\n",
str[2]  "          *****\n",
        "         *******\n",
        "        *********\n",
        "          *****\n",
        "         *******\n",
        "        *********\n",
        "       ***********\n",
        "      *************\n",
        "        *********\n",
        "       ***********\n",
        "      *************\n",
        "     ***************\n",
        "    *****************\n",
        "           ***\n",
        "           ***\n",
        "           ***\n"
    }; //二维字符数组的初始化
    int i;
    for(i=0; i<N; i++)
    {
        printf("%s", str[i]);
    }
    return 0;
}
```

```c
#include <stdio.h>
#define N 18
#define M 30
int main()
{
    char str[N][M] =
    {
        "            *",
        "           ***",
        "          *****",
        "         *******",
        "        *********",
        "          *****",
        "         *******",
        "        *********",
        "       ***********",
        "      *************",
        "        *********",
        "       ***********",
        "      *************",
        "     ***************",
        "    *****************",
        "           ***",
        "           ***",
        "           ***"
    }; //二维字符数组的初始化
    int i;
    for(i=0; i<N; i++)
    {
        printf("%s\n", str[i]);
    }
    return 0;
}
```

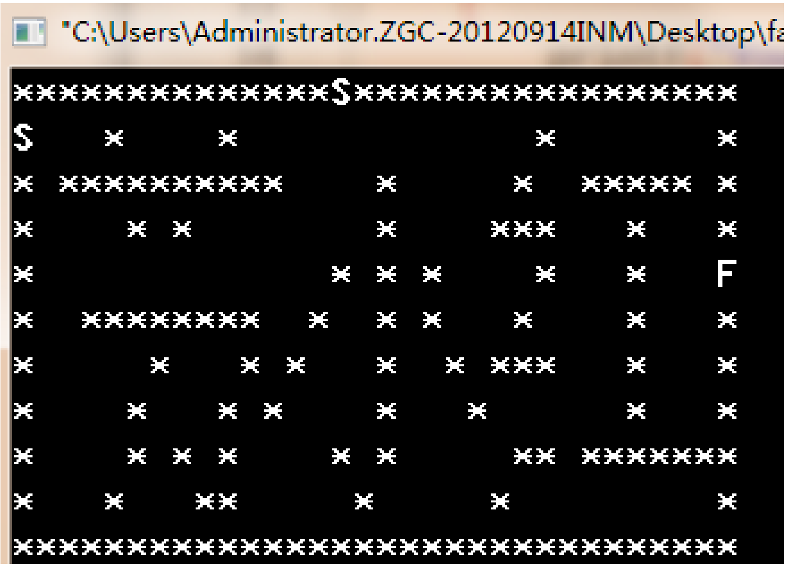
游戏时刻：如何存储一个迷宫？

```c
#include <stdio.h>
#define N 50
#define M 50
int main()
{
    char str[N][M] =
    {
        "************",
        "*o    *  ***",
        "* **  **   *",
        "*  *     * *",
        "* ****  ** *",
        "*        *  ",
        "************"
    };
    int i;
    for(i=0; i<7; i++)
    {
        printf("%s\n", str[i]);
        //puts(str[i]);
    }
    return 0;
}
```



"C:\Users\Administra

```
************
*o    *  ***
* **  **   *
*  *     * *
* ****  ** *
*        *
************
```

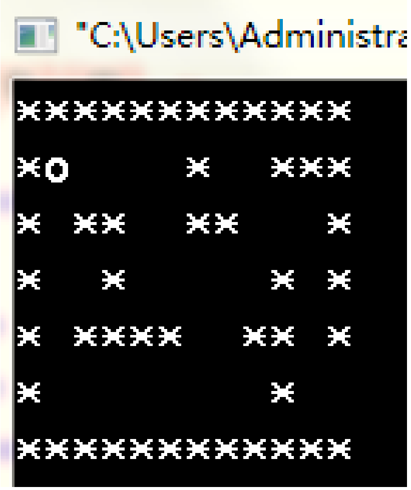


"C:\Users\Administrator.ZGC-20120914INM\Desktop\fa

```
**************S*****************
S   *    *             *       *
* **********    *     *  ***** *
*    * *        *    ***   *   *
*             * * *    *   *   F
*  ********  *  * *   *    *   *
*     *   * *   *  * ***   *   *
*    *   * *    *   *      *   *
*    * * *    * *     ** *******
*   *   **     *     *         *
********************************
```

游戏时刻：如何走迷宫？

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <windows.h>
#define N 50
#define M 50
int main()
{
    char str[N][M] =
    {
        "************",
        "*o    *  ***",
        "* **  **   *",
        "*  *     * *",
        "* ****  ** *",
        "*        *  ",
        "************"
    };
    Show(str);  //显示画面
    UpdateWithInput(str); //更新
    return 0;
}
```

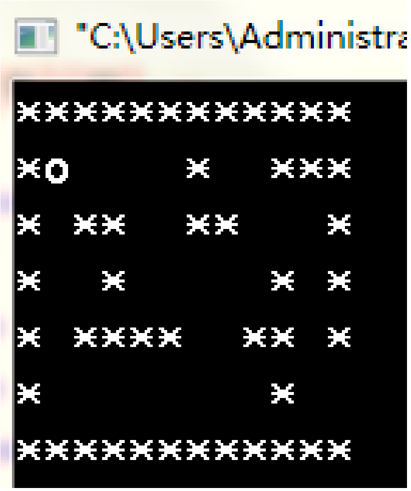


```
************
*o    *  ***
* **  **   *
*  *     * *
* ****  ** *
*        *
************
```

x-1,y-1	x-1,y	x-1,y+1
x,y-1	x,y	x,y+1
x+1,y-1	x+1,y	x+1,y+1

游戏时刻：如何走迷宫？

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <windows.h>
#define N 50
#define M 50
int main()
{
    char str[N][M] =
    {
        "************",
        "*o    *  ***",
        "* **  **   *",
        "*  *     * *",
        "* ****  ** *",
        "*        *  ",
        "************"
    };
    Show(str);  //显示画面
    UpdateWithInput(str); //更新
    return 0;
}
```

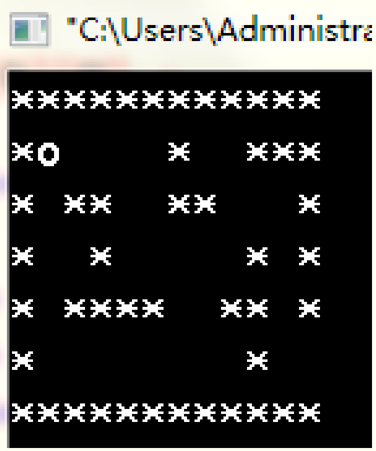


```
************
*o    *  ***
* **  **   *
*  *     * *
* ****  ** *
*        *
************
```

```
void Show(char str[][M])  //显示迷宫
{
    int i;
    for(i=0; i<7; i++)
    {
        puts(str[i]);
    }
}
```

```
void UpdateWithInput(char str[][M])     //与用户输入有关的更新
{
    int x = 1, y = 1;           //初始位置
    int exitX = 5, exitY = 12; //迷宫出口
    char input;
    while (x != exitX || y != exitY)
    {
        if (kbhit())            //判断是否有输入
        {
            input = getch();
            if (input == 'a' && str[x][y-1] != '*')   //左移
            {
                str[x][y] = ' ';
                y--;
                str[x][y] = 'o';
            }
            if (input == 'd' && str[x][y+1] != '*')    //右移
            {
                str[x][y] = ' ';
                y++;
                str[x][y] = 'o';
            }
            if (input == 'w' && str[x-1][y] != '*')   //上移
            {
                str[x][y] = ' ';
                x--;
                str[x][y] = 'o';
            }
            if (input == 's' && str[x+1][y] != '*')  //下移
            {
                str[x][y] = ' ';
                x++;
                str[x][y] = 'o';
            }
        }
        system("cls");
        Show(str);
        Sleep(200);
    }
    printf("You win!\n");
}
```

推箱子是一样的道理，你懂的

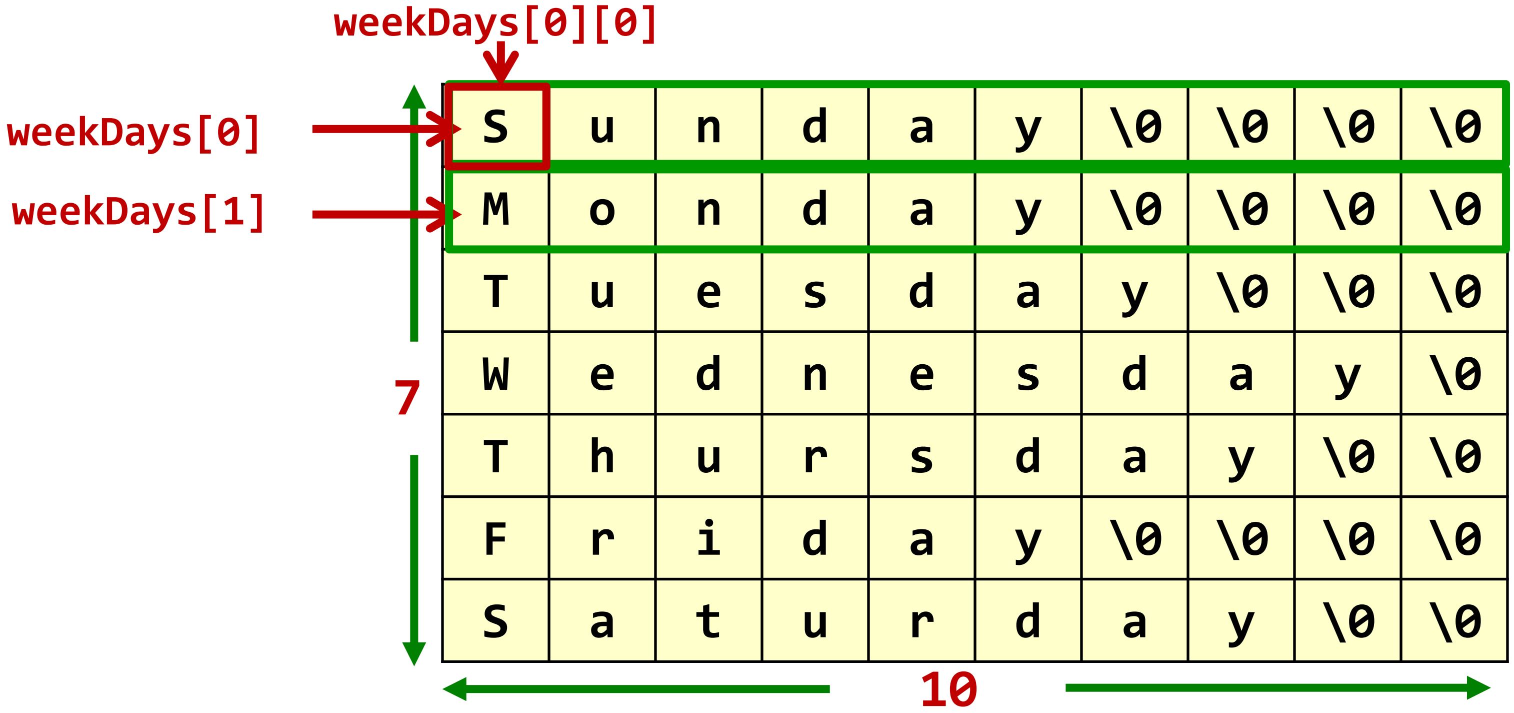


```
************
*o    *  ***
* **  **   *
*  *     * *
* ****  ** *
*        *
************
```

x-1,y-1	x-1,y	x-1,y+1
x,y-1	x,y	x,y+1
x+1,y-1	x+1,y	x+1,y+1

如何存储多行字符串？

```
char weekDays[][N] = {"Sunday", "Monday", "Tuesday", "Wednesday",
                      "Thursday", "Friday", "Saturday"};  //二维字符数组的初始化
```



S	u	n	d	a	y	\0	\0	\0	\0
M	o	n	d	a	y	\0	\0	\0	\0
T	u	e	s	d	a	y	\0	\0	\0
W	e	d	n	e	s	d	a	y	\0
T	h	u	r	s	d	a	y	\0	\0
F	r	i	d	a	y	\0	\0	\0	\0
S	a	t	u	r	d	a	y	\0	\0

```
for (i=0; i<7; i++)
{
    printf("%s\n", weekDays[i]);
}
```

```
for (i=0; i<7; i++)
{
    for (j=0; weekDays[i][j]!='\0'; j++)
    {
        printf("%c", weekDays[i][j]);
    }
    printf("\n");
}
```

如何对多个字符串排序？

- 按字典顺序对国名字符串排序

```
void SortString(char str[][N],int n)
{
   int   i, j;
   char  temp[N];
   for (i=0; i<n-1; i++)
   {
     for (j=i+1; j<n; j++)
     {
       if (strcmp(str[j],str[i])<0)
       {
          strcpy(temp, str[i]);
          strcpy(str[i], str[j]);
          strcpy(str[j], temp);
       }
     }
   }
}
```

A	m	e	r	i	c	a	\0	\0	\0
E	n	g	l	a	n	d	\0	\0	\0
A	u	s	t	r	a	l	i	a	\0
C	h	i	n	a	\0	\0	\0	\0	\0
F	i	n	l	a	n	d	\0	\0	\0

物理排序

A	m	e	r	i	c	a	\0	\0	\0
A	u	s	t	r	a	l	i	a	\0
C	h	i	n	a	\0	\0	\0	\0	\0
E	n	g	l	a	n	d	\0	\0	\0
F	i	n	l	a	n	d	\0	\0	\0

如何对多个字符串排序？

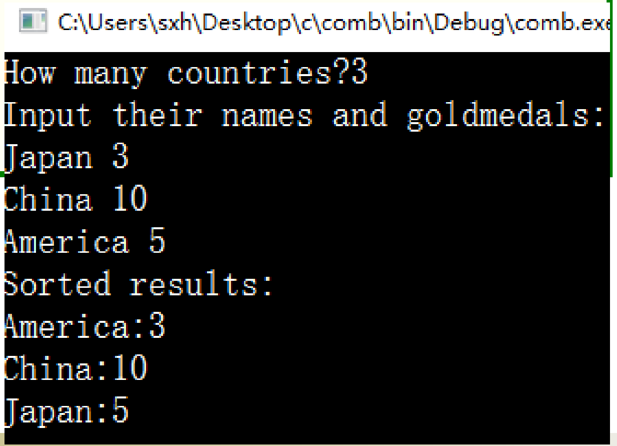
- 按字典顺序对国名字符串排序

```
void SortString(char str[][N],int n)
{
   int   i, j;
   char  temp[N];
   for (i=0; i<n-1; i++)
   {
     for (j=i+1; j<n; j++)
     {
        if (strcmp(str[j],str[i])<0)
        {
           strcpy(temp, str[i]);
           strcpy(str[i], str[j]);
           strcpy(str[j], temp);
        }
     }
   }
}
```

```
#include  <stdio.h>
#include  <string.h>
#define   M  150 //最多的字符串个数
#define   N  10  //每个字符串的最大长度
void SortString(char str[][N], int n);
int main()
{
   int    i, n;
   char   name[M][N];
   printf("How many countries?");
   scanf("%d",&n);
   getchar(); //读走输入缓冲区中的回车符
   printf("Input their names:\n");
   for (i=0; i<n; i++)
   {
       gets(name[i]);
   }
   SortString(name, n);
   printf("Sorted results:\n");
   for (i=0; i<n; i++)
   {
        puts(name[i]);
   }
   return 0;
}
```

如何对多条记录排序？

```
//按国名字典顺序排序
void SortString(char name[][N], int n)
{
   int   i, j;
   char  temp[N];
   for (i=0; i<n-1; i++)
   {
     for (j=i+1; j<n; j++)
     {
        if (strcmp(name[j], name[i]) < 0)
        {
           strcpy(temp, name[i]);
           strcpy(name[i], name[j]);
           strcpy(name[j], temp);
        }
     }
   }
}
```



C:\Users\sxh\Desktop\c\comb\bin\Debug\comb.exe

```
How many countries?3
Input their names and goldmedals:
Japan 3
China 10
America 5
Sorted results:
America:3
China:10
Japan:5
```

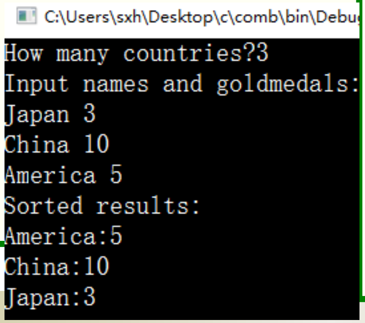
错在哪里？

```
#include  <stdio.h>
#include  <string.h>
#define   M  150 //最多的字符串个数
#define   N  10  //每个字符串的最大长度
void SortString(char name[][N], int n);
int main()
{
   int    i, n, goldmedals[M];
   char   name[M][N];
   printf("How many countries?");
   scanf("%d",&n);
   printf("Input names and goldmedals:\n");
   for (i=0; i<n; i++)
   {
      scanf("%s%d", name[i],
             &goldmedals[i]);
   }
   SortString(name, n);
   printf("Sorted results:\n");
   for (i=0; i<n; i++)
   {
      printf("%s:%d\n", name[i],
             goldmedals[i]);
   }
   return 0;
}
```

如何对多条记录排序？

```
//按国名字典顺序排序
void SortString(char name[][N], int goldmedals[],
                int n)
{
   int   i, j, t;
   char  temp[N];
   for (i=0; i<n-1; i++)
   {
     for (j=i+1; j<n; j++)
     {
        if (strcmp(name[j], name[i]) < 0)
        {
           strcpy(temp, name[i]);
           strcpy(name[i], name[j]);
           strcpy(name[j], temp);
           t = goldmedals[i];
           goldmedals[i] = goldmedals[j];
           goldmedals[j] = t;
        }
     }
   }
}
```

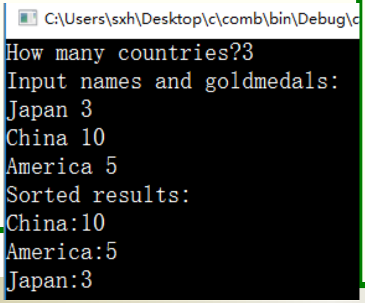
```
#include  <stdio.h>
#include  <string.h>
#define   M  150 //最多的字符串个数
#define   N  10  //每个字符串的最大长度
void SortString(char str[][N],
               int goldmedals[], int n);
int main()
{
   int    i, n, goldmedals[M];
   char   name[M][N];
   printf("How many countries?");
   scanf("%d",&n);
   printf("Input names and goldmedals:\n");
   for (i=0; i<n; i++)
   {
      scanf("%s%d", name[i],
               &goldmedals[i]);
   }
   SortString(name, goldmedals, n);
   printf("Sorted results:\n");
   for (i=0; i<n; i++)
   {
      printf("%s:%d\n", name[i],
               goldmedals[i]);
   }
   return 0;
}
```


C:\Users\sxh\Desktop\c\comb\bin\Debu

```
How many countries?3
Input names and goldmedals:
Japan 3
China 10
America 5
Sorted results:
America:5
China:10
Japan:3
```

如何对多条记录排序？

```
//按金牌数量降序排序
void SortString(char name[][N], int goldmedals[],
                int n)
{
   int   i, j, t;
   char  temp[N];
   for (i=0; i<n-1; i++)
   {
     for (j=i+1; j<n; j++)
     {
        if (goldmedals[j] > goldmedals[i])
        {
           strcpy(temp, name[i]);
           strcpy(name[i], name[j]);
           strcpy(name[j], temp);
           t = goldmedals[i];
           goldmedals[i] = goldmedals[j];
           goldmedals[j] = t;
        }
     }
   }
}
```



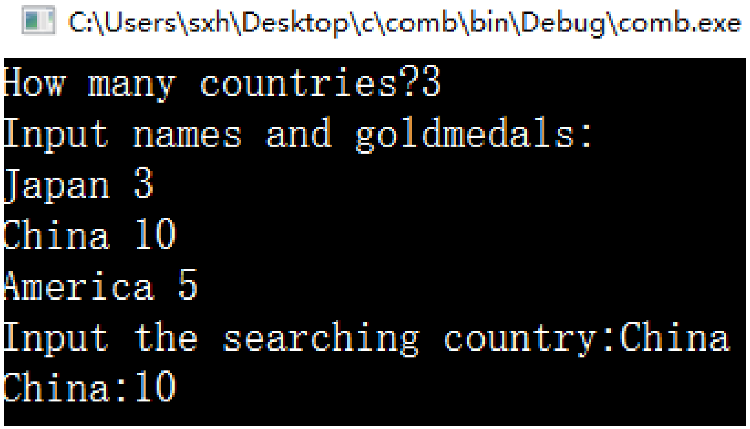
```
How many countries?3
Input names and goldmedals:
Japan 3
China 10
America 5
Sorted results:
China:10
America:5
Japan:3
```

```
#include  <stdio.h>
#include  <string.h>
#define   M  150 //最多的字符串个数
#define   N  10  //每个字符串的最大长度
void SortString(char str[][N],
              int goldmedals[], int n);
int main()
{
   int    i, n, goldmedals[M];
   char   name[M][N];
   printf("How many countries?");
   scanf("%d",&n);
   printf("Input names and goldmedals:\n");
   for (i=0; i<n; i++)
   {
      scanf("%s%d", name[i],
             &goldmedals[i]);
   }
   SortString(name, goldmedals, n);
   printf("Sorted results:\n");
   for (i=0; i<n; i++)
   {
      printf("%s:%d\n", name[i],
             goldmedals[i]);
   }
   return 0;
}
```

如何在多个字符串中查找指定的字符串？

```c
int main()
{
    int    i, n, pos, goldmedals[M];
    char   name[M][N], s[N];
    printf("How many countries?");
    scanf("%d",&n);
    printf("Input names and goldmedals:\n");
    for (i=0; i<n; i++)
    {
        scanf("%s%d", name[i],
              &goldmedals[i]);
    }
    printf("Input the searching country:");
    scanf("%s", s);
    pos = SearchString(name, n, s);
    if (pos != -1)
    {
        printf("%s:%d\n", s,
               goldmedals[pos]);
    }
    else
    {
        printf("Not found!\n");
    }
    return 0;
}
```

```c
//查找字符串dst在src多个字符串中的位置
int SearchString(char src[][N],int n,char dst[])
{
    int   i;
    for (i=0; i<n; i++)
    {
        if (strcmp(dst, src[i]) == 0)
        {
            return i; //返回下标位置
        }
    }
    return -1;  //表示没找到
}
```



C:\Users\sxh\Desktop\c\comb\bin\Debug\comb.exe

```
How many countries?3
Input names and goldmedals:
Japan 3
China 10
America 5
Input the searching country:China
China:10
```

考试攻略

- 考前务必把ppt上讲过的程序练会，争取多刷sse/mooc
- 下周考字符数组和结构体
  - 普通编程题，千万不要使用指针和结构体编写程序
  - 能用字符数组编写的程序，不要使用字符指针
  - 复杂编程题和附加改错题，可以使用指针和结构体编写程序
- 直接拷贝粘贴题目给的格式信息，避免输入输出格式错误
- 借助编译提示，尽量消灭程序中的语法错误，否则分数很低
- 熟记常用基本操作的算法和范例程序
- 不仅仅测试题目给定的测试用例，还要考虑边界条件的测试