



# 程序设计基础实验

The Basic Experiments of Programming Design

重庆工程学院通识学院

教师：王润生  
2022, Spring



# 课程安排



一、实验二代码展示

二、相关知识回顾

三、实验三“统计字符”

// 字符数组、指针



# 代码（某个不愿透露姓名的王同学）



```
1  #include<stdio.h>
2  #include<windows.h>
3  #include<conio.h>
4
5  #define ROW 9
6  #define COL 11
7  char map[ROW][COL+1]={
8  {"*#*****"},
9  {"***###*###*"},
10 {"####*#*****#"},
11 {"*#**###*#*"},
12 {"*****"},
13 {"#####*##*##"},
14 {"**#*****#*E"},
15 {"***#*###*#"},
16 {"*#*****"},
17 };
18
```



# 代码（某个不愿透露姓名的王同学）

```
18
19 void print_map()
20 {
21     for(int i=0;i<ROW;i++)
22     {
23         puts(map[i]);
24     }
25 }
26
27 void show_cursor(int x,int y)
28 {
29     COORD pos;
30     pos.X=x;
31     pos.Y=y;
32     printf("curX=%d,curY=%d\n",x,y);
33     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),pos);
34 }
35
```



# 代码（某个不愿透露姓名的王同学）

```
35
36 int curX,curY;
37 int main()
38 {
39     while(1)
40     {
41         system("cls");
42         print_map();
43         show_cursor(curX,curY);
44
45         char t=getch();
46         if(t=='w')
47         {
48             //向上走
49             if((curY-1)>=0 && (map[curY-1][curX]=='*' || map[curY-1][curX]=='E')) curY--;
50         }
51         else if(t=='s')
52         {
53             //向下走
54             if((curY+1)<ROW && (map[curY+1][curX]=='*' || map[curY+1][curX]=='E')) curY++;
55         }
56         else if(t=='a')
57         {
58             //向左走
59             if((curX-1)>=0 && (map[curY][curX-1]=='*' || map[curY][curX-1]=='E')) curX--;
60         }
61         else if(t=='d')
62         {
63             //向右走
64             if((curX+1)<COL && (map[curY][curX+1]=='*' || map[curY][curX+1]=='E')) curX++;
65         }
66
67         if(map[curY][curX]=='E')
68         {
69             break;
70         }
71     }
72
73     return 0;
74 }
75
76
```



重庆工程学院  
CHONGQING INSTITUTE OF ENGINEERING

# 课程安排



- 一、实验二代码展示
- 二、相关知识回顾
- 三、实验三“统计字符”

// 字符数组、指针



# 知识点回顾：字符数组



- 1、如何定义字符数组
- 2、字符数组初始化
- 3、怎么引用字符数组中的元素
- 4、字符串和字符串结束标志
- 5、字符数组的输入和输出
- 6、字符串处理函数



# 知识点回顾：字符数组



## 1、如何定义字符数组

字符型数据是以字符的**ASCII**码的形式存储的，一般占据一个字节。

**C语言中没有字符串类型，字符串存放在字符数组中。**

用来存放字符数据的数组就是字符数组。

```
char a; //定义一个字符变量a
```

```
char c[10]; //定义一个大小为10的字符数组c
```

```
char b[1000]; //定义一个大小为1000的字符数组b
```





# 知识点回顾：字符数组



## 2、字符数组初始化

对于字符数组的初始化，最容易理解的方式是使用“初始化列表”，把各个字符依次赋值给数组中的各个元素。

```
char c[10]={ 'l', ' ', 'a', 'm', ' ', 'h', 'a', 'p', 'p', 'y'};  
// 把10个字符依次赋值给c[0] ~ c[9]这10个元素
```

C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]	C[9]
'l'		'a'	'm'		'h'	'a'	'p'	'p'	'y'

如果在定义字符数组时不进行初始化，则数组中的各个元素的值都是不可预料的。如果花括号中的元素个数大于数组的长度，则会出现语法错误；如果花括号中的元素个数小于数组的长度，则剩余元素会被自动认定为空字符（即'\0'）

```
char c[]={ 'l', ' ', 'a', 'm', ' ', 'h', 'a', 'p', 'p', 'y'}; //初始化时可省略长度
```



# 知识点回顾：字符数组



## 3、怎么引用字符数组中的元素

可以引用字符数组中的元素，得到一个字符。

例：输出一个已知字符串。（先定义一个字符数组，并用“初始化列表”对其赋予初值，然后只用循环的方式逐一打印该字符数组中的字符）

```
1 #include<stdio.h>
2
3 int main()
4 {
5     char c[10] = {'I', ' ', 'a', 'm', ' ', 'h', 'a', 'p', 'p', 'y'};
6     for(int i=0;i<10;i++)
7     {
8         printf("%c",c[i]);
9     }
10    printf("\n");
11    return 0;
12 }
```

C:\Users\Administrator\Documents>  
I am happy  
Process returned 0 (0x0) execution time:  
Press any key to continue.



# 知识点回顾：字符数组



## 4、字符串和字符串结束标志

在C语言中，是将字符串作为字符数组来处理的。

```
char c[10]={'l',' ','a','m',' ','h','a','p','p','y'};  
//该例中字符串的实际长度与数组长度相等。
```

实际中，人们往往更关注字符串的有效长度而不是字符数组的长度，例如定义一个大小为1000的字符数组，而实际有效长度只有10。

```
char b[1000]={'l',' ','a','m',' ','h','a','p','p','y'};
```

C语言规定了一个字符串结束的标志，以空字符'\0'作为结束标志，如果一个字符数组有若干个字符，前面9个都不是空字符'\0'，而第10个字符是'\0'，则认为数组中有一个字符串，其有效字符为9个。

C语言用字符数组存放字符串常量时，会自动加'\0'作为结束



# 知识点回顾：字符数组



## 4、字符串和字符串结束标志

C语言用字符数组存放字符串常量时，会自动加'\0'作为结束

`char c[]="I am happy";` // 数组长度为11，因为末尾有个'\0'  
等价于

`char c[]={“I am happy”};` //数组长度为11  
等价于

`char c[]={‘I’,’,’,‘a’,‘m’,’,’,‘h’,‘a’,‘p’,‘p’,‘y’,‘\0’};` //数组长度为11

不等价于

`char c[]={‘I’,’,’,‘a’,‘m’,’,’,‘h’,‘a’,‘p’,‘p’,‘y’};` //数组长度为10

C语言用字符数组存放字符串常量时，会自动加'\0'作为结束



# 知识点回顾：字符数组



## 5、字符数组的输入和输出

char c[]="I like this course~";

字符数组的输出方式有两种：

- (1) 逐个字符输入输出，%c
- (2) 整个字符串一次输入输出，%s

```
1  #include<stdio.h>
2
3  int main()
4  {
5      char c[]="I like this course~";
6      //逐个字符输出
7      for(int i=0;i<sizeof(c);i++) printf("%c",c[i]);
8
9      printf("\n");
10
11     //整个字符串一次输出
12     printf("%s",c);
13
14     return 0;
15 }
```

C:\Users\Administrator\Do

I like this course~

I like this course~

Process returned 0 (0x0) exec

Press any key to continue.



# 知识点回顾：字符数组



## 6、字符串处理函数 `#include<string.h>`

C语言函数库提供了一些专门处理字符串的函数，十分方便。

`char c[]="where there is desire, there is gonna be a flame";`

`puts(c);`//将字符串（'\0'结束的字符序列）输出到终端

`gets(c);`//从终端输入一个字符串到字符数组

`strcat(c1,c2);`//将字符串c2拼接在字符串c1的后面

```
1  #include<stdio.h>
2
3  int main()
4  {
5      char c1[]="I like ";
6      char c2[]="this course~";
7      printf("%s",strcat(c1,c2));
8
9      return 0;
10 }
```

C:\Users\Administrator\Documents  
I like this course~  
Process returned 0 (0x0) execution time: 0.000 s  
Press any key to continue.



## 6、字符串处理函数

**strcpy(c1,c2);**//将字符串c2复制到c1中

**strncpy(c1,c2,n);**//将字符串c2中前n个复制到c1中去

**strcmp(c1,c2);**//比较字符串c1和c2，将两个字符串从左右到逐个字符相比（按照ASCII码值大小）

**strlen(c);**//测试字符串的长度，函数的值为字符串的实际长度（不包括'\0'）

**strlwr(c);**//将字符串中的大写字母变成小写

**strupr(c);**//将字符串中的小写字母变成大写

**#include<string.h>** //使用字符串函数时，在程序开头include



# 知识点回顾：指针



## **C Pointers**

In this tutorial, you'll learn about pointers; what pointers are, how do you use them.





重庆工程学院  
CHONGQING INSTITUTE OF ENGINEERING

# 地址：Address in C



还记得我们实验一定义的变量a,b,c吗？

```
int a, b, c;
```

&a就是变量a在内存中的地址！



# 地址：Address in C



其实，我们已经使用过地址很多次了！当我们使用scanf()这个函数的时候！

```
int a;
```

```
scanf("%d",&a);
```

//输入一个值存放在变量a的地址里

如：

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int a=5;
6      printf("a:%d\n",a);
7
8      printf("变量a的地址: %p",&a);
9      return 0;
10 }
11
```

```
a:5
变量a的地址: 0028ff0c
```



# 指针：C Pointers



Pointers(pointer variable) are special variables that are used to store addresses rather than values.

指针（指针变量）存放的是地址，而不是值。



# 指针语法：Pointer Syntax



我们可以如下声明指针： .

```
int* p;
```

```
Int * p1;
```

```
Int *p2;
```



# 指针：Pointer Syntax



我们可以如下赋值指针：

```
int* pc, c;
```

```
c = 5;
```

```
pc = &c;
```

5被赋值给变量c，  
c的地址被赋值给pc指针



# 指针：Pointer Syntax



我们可以如下获取指针指向的值： .

```
int* pc, c;
```

```
c = 5;
```

```
pc = &c;
```

```
printf("%d", *pc);
```

5被赋值给变量c，

c的地址被赋值给pc指针

获取指针指向的值，可以使用 **\*pc**



# 指针：Pointer Syntax



我们可以如下改变指针指向的值： .

```
int* pc, c;
```

```
c = 5;
```

```
pc = &c;
```

```
c = 1;
```

```
printf("%d", *pc); // output: 1
```

```
Printf("%d", c); // output: 1
```



# 指针 : Pointer Syntax



我们可以如下改变指针指向的值（Another）： .

```
int* pc, c;
```

```
c = 5;
```

```
pc = &c;
```

```
*pc = 1;
```

```
printf("%d", *pc); // output: 1
```

```
Printf("%d", c); // output: 1
```





# 指针：Pointer Syntax



我们可以如下改变指针指向的值（More）： .

```
int* pc, c, d;
```

```
c = 5;
```

```
d = -15;
```

```
pc = &c;
```

```
printf("%d", *pc); // output: 5
```

```
pc = &d;
```

```
printf("%d", *pc); // output: -15
```



# 指针挑战：Pointer Challenge



依次选4名同学逐一参与挑战

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int* pc, c;
6
7      c = 22;
8      printf("Address of c:%p\n", &c);
9      printf("Value of c:%d\n\n", c); // challenge 1?
10
11     pc = &c;
12     printf("Address of pointer pc:%p\n", pc);
13     printf("Content of pointer pc:%d\n\n", *pc); // challenge 2?
14
15     c = 11;
16     printf("Address of pointer pc:%p\n", pc);
17     printf("Content of pointer pc:%d\n\n", *pc); // challenge 3?
18
19     *pc = 2;
20     printf("Address of c:%p\n", &c);
21     printf("Value of c:%d\n\n", c); // challenge 4?
22
23     return 0;
24 }
```



# 指针挑战：Pointer Challenge



## 手敲验证

```
C:\Users\Administrator\Documents\test11\bin\Debu
Address of c:0028ff08
Value of c:22

Address of pointer pc:0028ff08
Content of pointer pc:22

Address of pointer pc:0028ff08
Content of pointer pc:11

Address of c:0028ff08
Value of c:2

Process returned 0 (0x0)   execution time : 0.109 s
Press any key to continue.
```



# Challenge 7

<https://pintia.cn/problem-sets/994805046380707840/problems/994805140211482624>

每个 PAT 考生在参加考试时都会被分配两个座位号，一个是试机座位，一个是考试座位。正常情况下，考生在入场时先得到试机座位号码，入座进入试机状态后，系统会显示该考生的考试座位号码，考试时考生需要换到考试座位就座。但有些考生迟到了，试机已经结束，他们只能拿着领到的试机座位号码求助于你，从后台查出他们的考试座位号码。

## 输入格式：

输入第一行给出一个正整数  $N$  ( $\leq 1000$ )，随后  $N$  行，每行给出一个考生的信息：`准考证号 试机座位号 考试座位号`。其中 `准考证号` 由 16 位数字组成，座位从 1 到  $N$  编号。输入保证每个人的准考证号都不同，并且任何时候都不会把两个人分配到同一个座位上。

考生信息之后，给出一个正整数  $M$  ( $\leq N$ )，随后一行中给出  $M$  个待查询的试机座位号码，以空格分隔。

## 输出格式：

对应每个需要查询的试机座位号码，在一行中输出对应考生的准考证号和考试座位号码，中间用 1 个空格分隔。

## 输入样例：

```
4
3310120150912233 2 4
3310120150912119 4 1
3310120150912126 1 3
3310120150912002 3 2
2
3 4
```

## 输出样例：

```
3310120150912002 2
3310120150912119 1
```



重庆工程学院  
CHONGQING INSTITUTE OF ENGINEERING

# 课程安排



- 一、实验二代码展示
- 二、相关知识回顾
- 三、实验三“统计字符”

// 字符数组、指针



# 实验：“统计字符”（背景



小王同学十分爱学习英语，一天晚上他在看快手小视频的时候看见了一句优美的句子：“It is not easy when you want to make something change, but if it is easy that everyone can make it”他突然想知道这句话里面有多少个it？于是他立即给他们班上会编程的同学打电话以寻求帮助。要求输入任意的一个英语句子和一串字符，请你帮小王同学输出该串字符在句子里面出现了多少次？



# 实验：“统计字符”（背景



输入：

Chinese is very simple in china

in

输出：

3

输入：

It is not easy when you want to make something  
change, but if it is easy that everyone can make it  
it

输出：

3



重庆工程学院  
CHONGQING INSTITUTE OF ENGINEERING

# 实验：“统计字符”（同学分析）



## 小组讨论分析







# 实验分析



- 1、句子输入
- 2、单词输入
- 3、如何存储
- 4、小大写？
- 5、如何匹配？



## 同学们编程实现“统计字符”

要求：

1. 理解并掌握该实验
2. 完成实验
3. 本次实验不用写实验报告
4. 上传代码到自己的代码仓库

(代码+流程图+代码仓库地址)



重庆工程学院  
CHONGQING INSTITUTE OF ENGINEERING

# The End



# Q&A