

# C语言第七章作业

## 输出阶乘表

【问题描述】5.2 生成一张阶乘表，输出1!~5!的值。要求定义和调用函数 `fact(n)` 计算  $n!$ ，函数类型为 `double`。

【输入输出样例】

```
1!=1
2!=2
3!=6
4!=24
5!=120
```

【样例说明】输出结束后需要换行。阶乘值输出格式为 `%.0f`。

## 解析

- 我们可以明确我们需要实现一个函数 `fact()`。
  - 这个函数的参数很明显是一个整形的数字。
  - 那么返回值呢？emmm，题目规定了，是一个 `double`

## 示例代码

```
1  #include <stdio.h>
2
3  //fact函数实现阶乘
4  double fact(int n){
5      double result = 1;
6      for (int i = 1; i <= n; i++)
7          result *= i;
8      return result;
9  }
10
11 //递归实现阶乘
12 double fact_1(int n){
13     if (n > 1)
14         return n * fact(n - 1);
15     if (n == 1 || n == 0)
16         return (double)1.0;
17     return (double)0.0;
18 }
19
20
21 int main()
22 {
```

```

23     //输出1到5的阶乘
24     for (int i = 1; i <= 5; i++)
25         printf("%d!=%.0f\n", i, fact(i));
26     return 0;
27 }

```

- 我这里使用了两种方法实现阶乘都可以使用

## 寻找水仙花数

【问题描述】5.3 输入2 个正整数m和n( $2 \leq m, n \leq 1000$ ), 输出m~n之间的所有水仙花数。水仙花数是指各位数字的立方和等于其自身的数。要求定义并调用函数 `is(number)` 判断number的各位数字之立方和是否等于其自身。

【输入输出样例】

Input m:100

Input n:400

153

370

371

【样例说明】输出结束后需要换行。输入提示符中冒号为英文字符，后面无空格。

## 解析

- 分析函数其实很清晰：
  - 参数：一个 `int` 型数据
  - 返回值： `false` 或者 `true` , 这里一般使用 `int` 来代替, 当然使用 `stdbool.h` 库会更加直观
  - 作用：判断一个数是否为水仙花数

## 示例代码

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  //自定义bool is(int n)判断n是否是水仙花数
5  bool is(int n){
6      int a,b,c;
7      a = n / 100;
8      b = n / 10 % 10;
9      c = n % 10;
10     if (a*a*a + b*b*b + c*c*c == n)
11         return true;
12     else
13         return false;
14 }

```

```

15
16 int main()
17 {
18     int n,m;
19     printf("Input m:");
20     scanf("%d",&m);
21     printf("Input n:");
22     scanf("%d",&n);
23     for (int i = m; i < n; i++)
24         if (is(i))
25             printf("%d\n",i);
26     return 0;
27 }

```

## 斐波拉契数列

【问题描述】5.4 输入2个正整数m和n( $m \geq 1$ ,  $n \leq 10000$ ), 输出m~n之间所有的Fibonacci数, 以整数输出。Fibonacci数列(第一项起): 1,1,2,3,5,8,13,21,...。要求定义并调用函数 **fib(n)**, 它的功能是返回第n项Fibonacci数。例如, **fib(7)** 的返回值是13。

【样例说明】输出结束后需要换行。输入提示符中冒号为英文字符, 后面无空格。

## 解析

- 我们只需要遍历所有的 **fibonacci** 数与输入的范围比较后输出就好了。

## 示例代码

```

1  #include <stdio.h>
2
3  //自定义函数fib(int n),返回第n项Fibonacci数
4  int fib(int n){
5      if (n == 1 || n == 2)
6          return 1;
7      return fib(n-1) + fib(n-2);
8      //利用递归完成求Fibonacci数
9  }
10
11 int main()
12 {
13     int n,m;
14     printf("Input m:");
15     scanf("%d",&m);
16     printf("Input n:");

```

```

17     scanf("%d",&n);
18
19     int i = 1;
20     while (1){
21         if (fib(i) >= m && fib(i) <= n)//如果第i项Fibonacci数大于
            等于m且小于等于n
22             printf("%d\n",fib(i));
23         if (fib(i) > n)//如果第i项Fibonacci数大于n
24             return 0;
25         i++;
26     }
27 }

```

## 统计数字2的个数

【问题描述】5.6 读入一个整数，统计并输出该数中2的个数。要求定义并调用函数 `countdigit(number,digit)`，它的功能是统计整数number中数字digit的个数。例如，`countdigit(10090,0)` 的返回值是3。

【输入输出样例】

Input an integer:21252

Number of digit 2:3

【样例说明】输出结束后不需要换行。输入提示符中冒号为英文字符，后面无空格。

## 解析

1. 我们首先需要把输入进来的每一位数字都分离开来计算 `digit = n%10; n /= 10;` 循环执行该语句即可

## 示例代码

```

1  #include <stdio.h>
2
3  //countdigit(number,digit), 它的功能是统计整数number中数字digit的个
    数。
4  int countdigit(int number,int digit){
5      int count=0;
6      while (number != 0){
7          if (number % 10 == digit)
8              count++;
9          number /= 10;
10     }
11     return count;
12 }
13
14 int main()

```

```

15 {
16     int n;
17     printf("Input an integer:");
18     scanf("%d",&n);
19
20     //输出2的个数
21     printf("Number of digit 2:%d\n",countdigit(n,2));
22
23     return 0;
24 }

```

## 数字杨辉三角

【问题描述】从键盘上输入整数n（设 $2 \leq n \leq 9$ ）按示例格式输出一个由数字组成的三角形图案。函数定义格式void szl(int n)。要求在主函数中输入整数n的值。在子函数中输出数字三角形图案。

【输入输出样例】

```

Enter n:5
      1
     1 2 1
    1 2 3 2 1
   1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1

```

【样例说明】输出结束后需要换行。输入提示符中冒号为英文字符，后面无空格。数值输出格式为%2d。

## 示例代码

```

1  #include <stdio.h>
2
3  //void szl(int n)实现输出n行杨辉三角
4  void szl(int n){
5      for (int i = 1; i <= n; i++){
6          for(int j = 1; j <= n-i; j++)
7              printf("%2c",' '); //补充空格使其居中
8          for (int j = 1; j <= i; j++)
9              printf("%2d",j); //输出第i行前半部分
10         for (int j = i-1; j >= 1; j--)
11             printf("%2d",j); //输出第i行后半部分
12         printf("\n");
13     }
14 }
15

```

```

16  int main()
17  {
18      int n;
19      printf("Enter n:");
20      scanf("%d",&n);
21      szt(n);
22      return 0;
23  }

```

## 输出日期

输入日期（年、月、日），输出它是该年的第几天。

## 解析

1. 初始化每一个月的天数，特殊的，判断年份是否是闰年
2. 将当前月份前的月份天数全部相加，本月的天数为日期的数字

## 示例代码

```

1  #include <stdio.h>
2
3  int main()
4  {
5      //输入日期（年、月、日），输出它是该年的第几天。
6      int year, month, day;
7      printf("Enter year month day:");//学习通上有点离谱month打成了
moth, 交作业的话看一下
8      scanf("%d%d%d", &year, &month, &day);
9
10     //判断是否为闰年
11     int days[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
30, 31}; //初始化每个月天数
12     if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
13         days[1] = 29; //如果是闰年，二月的天数加一
14
15     //计算该年的第几天
16     int sum = 0;
17     for (int i = 0; i < month - 1; i++)
18         sum += days[i]; //将month - 1个月的天数相加
19     sum += day; //将day相加
20     printf("day of year is %d", sum);

```

```
21     return 0;
22 }
```

## 查找指定元素

【问题描述】在数组中查找指定元素。输入一个正整数 $n$  ( $1 < n \leq 10$ )，然后输入 $n$ 个整数存入数组 $a$ 中，再输入一个整数 $x$ ，在数组 $a$ 中查找 $x$ ，如果找到则输出相应的最小下标，否则输出“Not Found”。要求定义并调用函数`search(list, n, x)`，它的功能是在数组`list`中查找元素 $x$ ，若找到则返回相应的最小下标，否则返回-1。

【输入输出样例1】

```
Input n:3
Input 3 integers:1 2 -6
Input x:2
index=1
```

【输入输出样例2】

```
Input n:5
Input 5 integers:1 2 3 4 5
Input x:0
Not Found
```

【样例说明】输出结束后需要换行。输入提示符中冒号为英文字符，后面无空格。

## 解析

- 这里就只是查找算法的一个应用，最简单的方式就是使用线性查找就行

## 示例代码

```
1  #include <stdio.h>
2
3  int search(int list[], int n, int x) {
4      for (int i = 0; i < n; i++) {
5          if (list[i] == x)
6              return i; //第一次找到就返回索引并退出
7      }
8      return -1; //如果未找到则返回-1
9  }
10
11 int main() {
12     int n;
```

```

13     printf("Input n:");
14     scanf("%d", &n);
15
16     int list[11];
17     printf("Input %d integers:", n);
18     for (int i = 0; i < n; i++)
19         scanf("%d", &list[i]);
20
21     int x;
22     printf("Input x:");
23     scanf("%d", &x);
24
25     int result = search(list, n, x);
26
27     if (result != -1)//如果找到
28         printf("index=%d", result);//输出索引
29     else
30         printf("Not Found");//如果未找到，输出"Not Found"
31
32     return 0;
33 }

```

## 寻找交集

【问题描述】输入两个由整型数构成的集合（元素个数均为5）分别放到数组A和B中，求这两个集合的交集（交集的元素由两个集合中的相同元素构成），最后输出交集集中的元素，要求在主函数中输入A和B数组，并输出交集集中的元素，在被调函数中实现求这两个集合的交集，返回交集元素的个数。定义函数为mix(a,b,c)，功能为将数组A和B中的交集赋给数组C，函数返回值为交集元素的个数。

【输入输出样例】

```

Please Enter Array A, 5 digits: 1 2 3 4 5
Please Enter Array B, 5 digits: 3 4 5 6 7
A B intersection is:   3   4   5

```

【样例说明】输出结束后不需要换行。输入提示符中冒号为英文字符，后面无空格。交集元素输出格式为 `%4d`。

## 解析

- 这个最简单的想法：
  - 先将两个数组都读入
  - 依次判断B数组中的数是否在A数组中出现
  - 如果出现就以 `%4d` 格式输出到屏幕上



## 示例代码

```
1  #include <stdio.h>
2
3  void mix(int a[], int b[]) {
4      for (int i = 0; i < 5; i++) {
5          for (int j = 0; j < 5; j++) {
6              if (a[i] == b[j]) {
7                  printf("%4d", a[i]); //找到交叉项后直接输出
8                  break;
9              }
10         }
11     }
12 }
13
14 int main() {
15     int A[5], B[5];
16
17     printf("Please Enter Array A,5 digits:");
18     for (int i = 0; i < 5; i++)
19         scanf("%d", &A[i]);
20
21     printf("Please Enter Array B,5 digits:");
22     for (int i = 0; i < 5; i++)
23         scanf("%d", &B[i]);
24
25     printf("A B intersection is:");
26     mix(A, B);
27     return 0;
28 }
```

## 字符串的定位复制

【问题描述】编写程序，将字符串中的第m个字符开始的全部字符复制到另一个字符串中。要求在主函数中输入字符串及m的值，并输出复制结果，在被调函数中完成复制。

【输入形式】输入一个字符串及整数m。

【输出形式】输出复制后的结果，输出后换行。

【样例输入】

Enter a string: shanghai

Enter m: 6

【样例输出】

hai

#### 【样例说明】

输入提示符后要加一个空格。例如“Input integers: ”, 其中“:”后要加一个且只能一个空格。输出后换行。

英文字母区分大小写。必须严格按样例输入输出。

## 解析

- 这里需要复制字符串，在不使用指针的情况下只能赋值。

## 示例代码1

```
1  #include <stdio.h>
2
3  void copySubstring(char* str, int m, char* newStr) {
4      int i, j;
5
6      // 复制从第m个字符开始的字符串
7      for (i = m - 1, j = 0; str[i] != '\0'; i++, j++)
8          newStr[j] = str[i];
9
10     newStr[j] = '\0'; //手动添加字符串结束符
11 }
12
13 int main() {
14     char str[100], cstr[100];
15     int m;
16
17     printf("Enter a string: ");
18     gets(str); //这里不建议使用scanf("%s", str);可能会遇到含有空格的情况
19
20     printf("Enter m: ");
21     scanf("%d", &m);
22
23
24     copySubstring(str, m, cstr);
25
26     printf("%s\n", cstr);
27
28     return 0;
29 }
```

## 示例代码2

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void copystr(char *s1, char *s2, int n) {
5      char *p;
6      p = s1 + n - 1;
7      strcpy(s2, p);
8  }
9
10 int main() {
11     char str[100], cstr[100];
12     int m;
13
14     printf("Enter a string: ");
15     gets(str); //这里不建议使用scanf("%s", str);可能会遇到含有空格的情况
16
17     printf("Enter m: ");
18     scanf("%d", &m);
19
20     copystr(str, cstr, m);
21
22     printf("%s\n", cstr);
23
24     return 0;
25 }
```

- 这里合理的运用库函数和指针可以使代码简洁很多

## 进制转换

【问题描述】输入一个十进制整数和想转换的进制数，利用函数将其转换成为二进制、八进制或十六进制数（0~9、A-F注意为大写）

【样例输入1】 15 2

【样例输出1】 the converted the number is:1111

【样例输入2】 27 16

【样例输出2】 the converted the number is:1B

【样例说明】把15转换为2进制是1111，输出后换行；把27转成16进制是1B，输出后换行

## 解析

- 短除法还是库函数都是ok的

## 示例代码

```
1  #include<stdio.h>
2
3  void decimalToBinary(int decimal) {
4      int binary[32];
5      int i = 0;
6
7      while (decimal > 0) {
8          binary[i] = decimal % 2;
9          decimal = decimal / 2;
10         i++;
11     }
12
13     printf("the converted the number is:");
14
15     for (int j = i - 1; j >= 0; j--) {
16         printf("%d", binary[j]);
17     }
18
19     printf("\n");
20 }
21
22 int main()
23 {
24     int n,d;
25     scanf("%d%d",&n,&d);
26     switch (d)
27     {
28     case 2:
29         //将10进制的n转化为2进制输出
30         decimalToBinary(n);
31         /*
32         这里是因为有一个非标准库函数无法使用要不然也很简单:
33         #include <stdlib.h>
34         char str[32];
35         itoa(n, str, 2);
36         printf("the converted the number is:%s\n", str);
37         这样就十分的简洁明了, 可惜不能用
38         */
39         break;
40     case 8:
41         //将10进制的n转化为8进制输出
42         printf("the converted the number is:%o",n);
43         break;
44     case 16:
```

```
45         //将10进制的n转化为16进制输出
46         printf("the converted the number is:%X",n);
47         break;
48     default:
49         break;
50     }
51     return 0;
52 }
```