

C语言第五章作业-3

输出对应的星期

【问题描述】用整数1~7依次表示星期一至星期日。由键盘输入一个整数，输出对应的英文表示，如果输入的整数在 1~7 之外，输出“Error”信息。

【样例输入1】 1

【样例输出1】 Monday

【样例输入2】 7

【样例输出2】 Sunday

【样例输入3】 10

【样例输出3】 Error

【样例说明】 本题输出结束后无换行符。

解析

- 这里很明显应该使用 `switch(){}` 条件选择语句
 - 使用嵌套的 `if` 会显得自己很蠢，
 - `if` 的嵌套在超过3层以后，效率会开始低于 `switch`
- 将输入的数字作为 `switch` 的判断参数就好了。

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int day;
4      scanf("%d", &day); //简单的使用switch()语句进行判断的示例
5      switch (day) {
6          case 1:
7              printf("Monday");
8              break;
9          case 2:
10             printf("Tuesday");
11             break;
12          case 3:
13             printf("Wednesday");
14             break;
15          case 4:
16             printf("Thursday");
17             break;
18          case 5:
19             printf("Friday");
20             break;
```

```

21         case 6:
22             printf("Saturday");
23             break;
24         case 7:
25             printf("Sunday");
26             break;
27         default:
28             printf("Error");
29             break;
30     }//记得每一个case和default后面要加break
31     return 0;
32 }

```

三角形类型判断

【问题描述】从键盘输入三角形的三个边长判断能否构成三角形，如能构成三角形，则判断是哪一种类型（等腰三角形 **isosceles triangle**、等边三角形 **equilateral triangle**、直角三角形 **right triangle**、任意三角形 **arbitrary triangle**）；否则输出“**not triangle**”。

【输入形式】输入三个双精度浮点数

【输出形式】判断输出语句结束后跟换行符 **\n**，所有输出语句字符小写。

【样例输入1】 **3.0 3.0 4.0**

【样例输出1】 **isosceles triangle**

【样例输入2】 **3.0 3.0 3.0**

【样例输出2】 **equilateral triangle**

【样例输入3】 **1.0 3.0 6.0**

【样例输出3】 **not triangle**

解析

- 我们这里主要需要确定判断的顺序：
 - 判断是否是三角形
 - 判断是否是等边三角形
 - 判断是否是等腰三角形
 - 判断是否是直角三角形
 - 判断是否为其他三角形

示例代码

```
1  #include <stdio.h>
2  int main() {
3      double a, b, c;
4      scanf("%lf %lf %lf", &a, &b, &c); //输入三边
5
6      if (a + b > c && a + c > b && b + c > a) { //判断是否为三角形
7          if (a == b && b == c) { //判断是否为等边三角形
8              printf("equilateral triangle\n");
9          } else if (a == b || b == c || a == c) { //判断是否为等腰
            三角形
10             printf("isosceles triangle\n");
11         } else if (a * a + b * b == c * c || a * a + c * c ==
            b * b || b * b + c * c == a * a) { //判断是否为直角三角形
12             printf("right triangle\n");
13         } else {
14             printf("arbitrary triangle\n"); //判断是否为其他三角形
15         }
16     } else {
17         printf("not triangle\n"); //输出不是三角形
18     }
19     return 0;
20 }
```

- 这只是很简单的写一个判断问题，只需要分析好判断顺序就是ok的。

猴子吃桃

【问题描述】猴子吃桃。猴子第一天摘下若干桃子，当即吃了一半，还不过瘾，又多吃了一个，第二天早上将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半再加一个桃子。到第n天，只剩下一个桃子。求第一天共摘下多少桃子。

【输入形式】输入剩下一个桃子的天数

【输出形式】输出桃子的数量，输出结束有换行符。

【样例输入1】 6

【样例输出1】 94

【样例输入2】 10

【样例输出2】 1534

解析

- 这道题我个人还是建议倒着来运算，即从最后一天剩下一个桃子的时候开始计算
 - 我们根据猴子吃桃的规则分析， **后一天 = 前一天/2 - 1**
 - 因为我们倒着运算，所以可以得到公式： **前一天 = 2 * (后一天 + 1)**
 - 那么一共吃了n天，我们只需要从 **后一天 = 1** 开始计算n次就得到了我们的结果

示例代码

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     scanf("%d", &n); //输入天数n
5
6     int peach = 1; //初始化桃子个数
7     for (int i = n; i >= 2; i--)
8         peach = (peach + 1) * 2; //每天的桃子个数=上一天的桃子个数/2
9     printf("%d\n", peach);
10    return 0;
11 }
```

- 这里就是一个倒推思想的应用，简单的循环 `for` 的应用。

斐波拉契数列的输出

【问题描述】求1~n项 **Fibonacci** 数列的数值。

【输入形式】读入正整数值n项的值。

【输出形式】每行输出5个数值，每个数值占8位。输出结束有换行符。

【补充说明】数列的前两项为1, 1

解析

- 我们可以知道斐波拉契数列的递推公式为： $a_{n+2} = a_{n+1} + a_n$ ，题目给出的初始数据为： $a_1 = 1, a_2 = 1$
- 输出限制：
 - 单个输出的输出宽度为8，我们可以使用 `%8d` 来控制这个输出宽度
 - 每输出5个值需要换行所以我们需要一个计数器每一次计满5次就输出一个 `\n`

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int n;
4      scanf("%d", &n);
5      int a = 1, b = 1; //设置Fibonacci数列的前两项
6      int count = 0;
7      while (count < n) {
8          printf("%8d", a); //按8字符宽度输出Fibonacci数列第n项
9          count++; //计数
10         if (count % 5 == 0) //每5项换行
11             printf("\n");
12         int temp = a + b; //计算下n+2项
13         a = b; //将n+1项赋值给n项
14         b = temp; //将n+2项赋值给n+1项
15     }
16     if (count % 5 != 0)
17         printf("\n"); //在完成最后输出后如果不满5个则补上一个换行
18     return 0;
19 }
```

买鸡问题

【问题描述】百钱买百鸡。“鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。百钱买百鸡，问鸡翁、母、雏各几何？”

【输出形式】输出所有的解，每行输出一组解。输出结束有换行符。

【样例输出】

cock=0,hen=25,chicken=75

cock=4,hen=18,chicken=78

cock=8,hen=11,chicken=81

cock=12,hen=4,chicken=84

解析

- 假设买a只鸡翁，b只鸡母，c只鸡仔，需要满足的表达式为： $5a + 3b + c = 100$ 和 $a + b + c = 100$
- 我们只要求出上面表达式在定义域范围内的所有解就可以了
 - 根据第二个表达式我们可以将第一个表示转化为一个二元一次方程。
 - 定义域 $a \leq 20, b \leq 33$ 。
 - 我们遍历定义域内所有点就行了。

```

1  #include <stdio.h>
2  int main() {
3      int x, y, z;
4      for (x = 0; x <= 20; x++) { //遍历0~20只公鸡
5          for (y = 0; y <= 33; y++) { //遍历0~33只母鸡
6              z = 100 - x - y; //计算小鸡数量
7              if (5 * x + 3 * y + z / 3 == 100 && z % 3 == 0) //总
                金额为100元时输出结果
8                  printf("cock=%d,hen=%d,chicken=%d\n", x, y,
9                      z);
10             }
11         }
12     }
13     return 0;
14 }

```

寻找阿姆斯特朗数（水仙花数、自幂数）

【问题描述】一个数如果等于其每一个数字立方之和，则此数称为阿姆斯特朗数。如407就是一个阿姆斯特朗数，因为： $407 = 4^3 + 0^3 + 7^3$ 。要求输出100~999间所有的阿姆斯特朗数。

【输出形式】要求输出100~999间所有的阿姆斯特朗数。每个数值占6位。输出结束无换行符。

【样例输出】 153 370 371 407

解析

- 我们需要将3位数的每一位的数字都分解出来进行立方并相加
 - 通常我们使用 `a%10` 来获取 `a` 的最低位数字
 - 使用完成后我们使用 `a = a/10` 来处理掉最后一位数字，重复上一步得到倒数第二位数字
 - 一般我们将 `a = a/10` 简写为 `a /= 10`
- 计算一个数的立方我们有两种方案：
 - 直接书写 `a * a * a`，手动书写运算式
 - 使用 `pow(x,y)` 函数计算x的y次方，`pow((double)a,(double)3.0)`
 - 两张写法都是OK的

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int num, digit, sum;
4      for (num = 100; num <= 999; num++) { //遍历100-999中所有数
5          sum = 0;
6          int temp = num;
7          while (temp > 0) {
8              digit = temp % 10;
9              sum += digit * digit * digit;
10             temp /= 10;
11         } //计算每一位数字的立方和
12         if (sum == num) //判断是否是阿姆斯特朗数(水仙花数)
13             printf("%6d", num);
14     }
15     return 0;
16 }
```

寻找完数

【问题描述】 一个数如果恰好等于它的因子之和，这个数就称为“完数”。求1000以内的所有完数。

【输出形式】 1000以内的所有完数。每个数值占6位。输出结束无换行符。

【样例输出】 6 28 496

【样例说明】 完数 6=1+2+3

解析

- 我们首先需要遍历 1~1000 以内的所有整数进行判断
- 我们要寻找n的所有因数：
 - 遍历 1~n
 - 满足条件为 $n \% i == 0$ 则i是n的因数
 - n本身不算是自己的因数所以遍历应该是 1~(n-1)

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int num, sum;
4      for (num = 1; num <= 1000; num++) { //遍历1-1000中所有数
5          sum = 0;
6          for (int i = 1; i < num; i++) {
7              if (num % i == 0)
8                  sum += i;
9          } //计算其因子的和
10         if (sum == num) //判断是否是完数
11             printf("%6d", num);
12     }
13     return 0;
14 }
```

- 代码优化提示：其实我们可以知道若 $x > n/2$ 且 $x \neq n$ 则 $n \% x \neq 0$ 所以我们只需要遍历 $1 \sim (n/2)$

分解正整数的质因数

【问题描述】将一个正整数分解质因素。例如，输入90，输出90=2*3*3*5。

【输入形式】从键盘输入一个整数。

【输入输出样例1】 **Input an integer:90 90=2*3*3*5**

【输入输出样例2】 **Input an integer:18 18=2*3*3**

【样例说明】输入提示符后无空格。输出语句的“=”两边无空格。输出结束无换行符。英文字母区分大小写。必须严格按样例输入输出。

解析

1. 如果整数能够被当前尝试的数字整除，则该数字是一个质因数。
2. 打印出该质因数，并将输入的整数除以该质因数，更新输入的整数为除法运算的结果。
3. 重复步骤 3 和 4，直到输入的整数无法再被除尽，即输入的整数等于 1。

示例代码

```
1 #include <stdio.h>
2 int main() {
3     int num, i;
4     printf("Input an integer:");
5     scanf("%d", &num); //输入一个整数
6     printf("%d=", num);
7     for (i = 2; i <= num; i++) {
8         while (num % i == 0) { //判断是否是质因数
9             printf("%d", i);
10            num /= i;
11            if (num != 1)
12                printf("*");
13        }
14    }
15    return 0;
16 }
```

- 解释什么是质因数：
 - 因为我们把后面得到的那个因数在不断的分解直至最后那个因数无法分解，那么它一定是质数。

输出*号菱形

【问题描述】根据用户输入的n，输出边长为n的*号菱形

【输出说明】输出结束有换行符。

解析

- 最长的一行输出为 $2n-1$ 个星号，往上往下依次减去两个星号直至为1
 - 注意每次其实都是输出 $2n-1$ 个字符，只是非星号部分用空格代替
 - 但是在不足 $2n-1$ 个星号的时候，后续的位置不需要输出空格直接换行就好

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int n, i, j;
4      scanf("%d", &n);
5      // 打印上半部分
6      for (i = 1; i <= n; i++) {
7          for (j = 1; j <= n - i; j++)
8              printf(" ");
9          for (j = 1; j <= 2 * i - 1; j++)
10             printf("*");
11         printf("\n");
12     }
13
14     // 打印下半部分
15     for (i = n - 1; i >= 1; i--) {
16         for (j = 1; j <= n - i; j++)
17             printf(" ");
18         for (j = 1; j <= 2 * i - 1; j++)
19             printf("*");
20         printf("\n");
21     }
22     return 0;
23 }
```

空心星号三角形

【问题描述】根据用户输入的n，输出边长为n的*号空心等边三角形

【样例输入】5

【输出说明】输出结束有换行符。

解析

- 这个输出的基础和上一个代码输出下半部分的输出是大差不错的，修改的地方只有两个
 - 输出第一行的时候将偶数次的输出更换为空格
 - 输出后面几行的时候除了开头和结尾的地方都换为空格

示例代码

```
1  #include <stdio.h>
2  int main() {
3      int n, i, j;
4      scanf("%d", &n);
5
6      for (i = n; i >= 1; i--) {
7          for (j = 1; j <= n - i; j++)
8              printf(" ");
9          if (i == n || i == 1) {
10             for (j = 1; j <= 2 * i - 1; j++) {
11                 if(j % 2 != 0)
12                     printf("*");
13                 else
14                     printf(" ");
15             }
16         }
17         else {
18             printf("*");
19             for (j = 1; j <= 2 * i - 3; j++)
20                 printf(" ");
21             printf("*");
22         }
23         printf("\n");
24     }
25
26     return 0;
27 }
```