北京林业大学 2020-2021 学年第 1 学期

程序设计基础 实验任务书 2

实验题目：循环与分支

实验环境：Windows 10, Visual Studio 2015/2019

实验目的

1. 学习和使用循环语句 while, for, do/while；
2. 学习和使用分支语句 if, if/else。
3. 循环嵌套。
4. 循环和分支嵌套

实验内容：

1. 熟悉 while 语句。P118(6.2)。

   while(expression) statement

   其中 expression 是表示真假的表达式。如果表达式的值为真，则执行语句 statement。

   - 有关真假的表达方式。
     - _Bool 类型。1 真，0 假。6.3.4 (程序 6.9)
     - 关系表达式。比较数值大小关系，结果是真假，6.3 (表 6.1)。例如 scanf 函数成功时返回 1，失败时返回 0。就可以使用类似 scanf("%d", &num)==1 表示是否成功的真假值。
     - 整数。0 为假，非 0 为真。6.3.2 (程序 6.7)。例如 scanf 函数成功时返回 1，失败时返回 0。就可以使用类似 scanf("%d", &num)表示是否成功的真假值。
     - 逻辑表达式。对真假的结果进行逻辑运算。
   - 有关循环
     - 循环初始化。如果用变量来控制循环，需要对该变量赋合适的初值。
     - 循环条件。需要设置合适的条件（通过计算真假的表达式），以控制循环。
     - 退出循环。如果用变量来控制循环，则需要改变变量的值，以结束循环。

   输入 n，使用 while 循环，计算 1 到 n 以内的整数的和。

   （注：参考程序 6.1。其要点是循环里使用 sum = sum+num; 这种方式 sum 加一个值，结果然后放 sum，称为累加。）

2. 继续熟悉 while 语句。

   输入 n，使用 while 循环，计算 2 的 n 次幂。

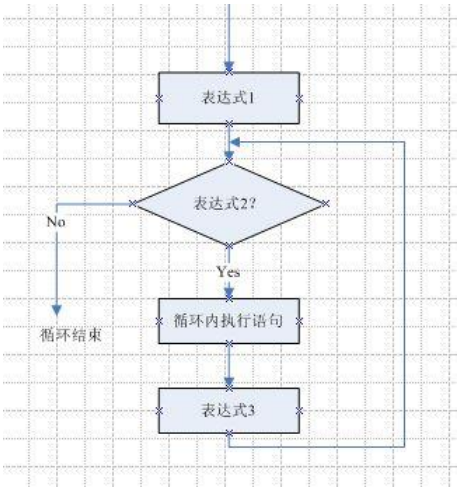
   （累乘）

3. 熟悉 for 循环。程序 6.11

   for (nitialize(初始化，表达式 1; test(检验，表达式 2); update(更新，表达式 3))

   statement(循环内执行语句)

下一条语句



for 语句具有很好的灵活性。6.5.1 (利用 for 的灵活性)。大概了解 for 循环的多种例子。
输入 n，使用 for 循环，计算 2 到 n 以内的偶数的和。
（注：参考 6.5.1，利用 for 的灵活性-2）

4. 熟悉 if 语句。

   if(expression) statement

   如果 expression 求得的值为真（非零），就执行 statement；否则，跳过该语句。

   if (expression) statement1

   else statement2

   如果 expression 为真（非零），就执行 statement1；如果 expression 为假或零，则执行跟在 else 后的那一条语句 （statement2）。

   输入整数 n,k，输出 k 是否能整除 n 的结果。
   （注：此处结果为真假。先在程序中用一个变量来保存这个真假值，然后根据这个变量的真假，输出文字信息。
   整除的表达不会？不应该的，实在不会，随便问周围同学好了
   ）

5. 循环和分支的嵌套。（循环里有 if）
   表达多个条件时，需要使用逻辑表达式
   编写一个程序,统计并输出能被 3 整除或者能被 5 整除或者能被 7 整数的所有 3 位整数，按每行 7 个输出。
   （注：3 位数怎么表达？100-999 不就是?
   每行 7 个输出，其实就是每输出 7 个数，加一个换行。于是每输出一个数的同时，要记一下数，当该数能整除 7 的时候，就 printf(“\n”);
   ）

6. 嵌套循环。(循环里包含循环，程序 6.17)
   输出如下图形(**‘*’就是’* ’外加一个空格**)。

   ＊
   ＊＊＊

  ＊＊＊＊＊
＊＊＊＊＊＊＊
  ＊＊＊＊＊
   ＊＊＊
     ＊
    (注：先用一个两重循环实现上半部分，然后再下半部分＊
    找规律。”＊ “的个数 1，3，4，7，对应的空格个数 6，4，3，0。
    循环应该是类似下面这样
    for(i=1; i<=7; i+=2)
    {
        k = 7-i;
        画 k 个空格
        画 i 个”＊ “
        换行
    }
    至于画 i 个”＊ “怎么做，简单一个 for 循环就可以了
)

7. 让程序要求用户输入一个字母，使用嵌套循环产生像下面这样的大写字母金字塔图案：
    A
   ABA
  ABCBA
 ABCDCBA
ABCDEDCBA
这种图案要扩展到用户输入的字符。例如，前面的图案是在输入 e 或 E 时需要产生的。
(注：有一种简单做法，char chs[27]=”ABCDE”;输出 ABCDE 就变成输出数组 chs 的第 0 到第 4 个元素。
然后找规律。每行输出字符个数 1，3，5，7，9。
如果某行输出 2*k+1 个字符，相当于输出数组的第 0…k-1，k，k-1,….0 个元素
)

8. 数组。P140
    定义一个大小为 10 整数数组 arry，给数组的每个元素赋值。输出这个数组各个的值。
    计算数组元素的最大值。
    （给数组的每个元素赋值：大概就是循环 ，scanf(“%d”，&arry[i]，参考 P141，程序 6.19
    求最大值的一种做法：定义最大值 maxValue，先让它等于第一个元素的值，然后和第二个元素比较，if(maxValue<arry[1]) maxValue=arry[1]; 这样循环一遍就可以了
    ）

9. 判定给定的一个数 n 是否为素数。

    循环(k 从 2 到 n-1)
    {

if k|n

退出循环

}

根据 k 的值，判定是否为素数(如果所有的 k 不能整除 n，那么退出循环时 k 的值为 n 了，是不是这样？)

（提示，可以用 for(k=2; k<n; k++)来循环，也可以 while(k++<n）来做从 2 到 n-1 的循环)

10.选择排序

【背景描述】：选择排序(Selection sort)是一种简单直观的排序算法。它的工作原理如下。首先在未排序序列中找到最小（大）元素，存放到排序序列的起始位置，然后，再从剩余未排序元素中继续寻找最小（大）元素，然后放到已排序序列的末尾。以此类推，直到所有元素均排序完毕。

(提示：

a) 在未排序序列（数组的 k1 位置到 k2 位置）中找到最小元素

类似这样的

```
int minV=arry[k1];
int index = k1; //用来记录最小值的位置
for(k=k1; k<k2; k++)
{
  minV 和 arry[k]比较，如果 minV 更大，更新 minV 的值和 index 的值
}
```

b) 存放到排序序列的起始位置

就是把找出来的最小值和第一个数交换位置。类似这样的，相当于两个人换位置，一个人往旁边空位上移动一下，然后继续交换

```
int temp = arry[k1];
arry[k1] =  arry[index];
arry[index] = temp;
```

)

【输入】：第一行一个整数 n，表示待排序的数的个数，2<=n<=1000。第二行 n 个待排序的整数，由空格隔开，范围为-2^{31}到$2^{31}-1$

【输出】：n 个数，按从小到大排序

【输入样例】：

5

4 2 3 1 5

【输出样例】：

1 2 3 4 5

【思考】：当 n 等于 10000 或者 n 等于 100000 时，选择排序要执行多长时间，有没有更好的排序方法。