序 号 41 学 号 20211120138 姓名 薛凡豪 成绩

试验次数 04 实验名称程序函数的使用及应用 日期11.11

# 任务一 日期和天数的转换

## 1.1伪代码

1.设置菜单选项，即满足用户选择date\_num or num\_date

2.完成自定义函数date\_num 和num\_date的编写

3.调用函数并且让用户输入相应的值

（由于此次任务相对复杂，所以这里给出了简要的流程说明，并没有给出具体的实现过程）

## 1.2程序的运行测试

（1）由于用户可能需要重复输入，这里为了简便，我运用了cls清屏功能，和do while循环结构来达到重复运行程序的目的。（但是有个小缺憾，因为do while是至少进行一次循环的，所以刚开始进入按退出按钮是不能直接退出对话框的）

（2）这里我给出了2021年（平年）和2020年（闰年）的date\_num以及num\_date相同日期的分别测试，既能纵向对比，也能横向对比。使得结果更加准确。

（3）相应的源代码见**附录一**

下面给出2021年11月11日的运行结果。

图片包含 文本

描述已自动生成

文本

描述已自动生成

下面给出2020年11月11日的运行结果

图形用户界面, 文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

（4）很显然2020年作为闰年和2021年作为平年的2月是相差一天的，我们的结果给出了很好的证明，这证明我们的2月代码是运行无误的。

（5）但是其实这里我还做出了更多的测试，像百年整除的闰年判定测试，像1月是否运行正确的测试等。但这里为了节约空间并没有再赘述。但是**一个较为复杂的程序还是需要严谨的运行测试才可以作为成果来使用，这是我们需要谨记的**。.

## 1.3总结

1.对于较复杂的程序可以分解问题来实现，可以分步骤完成，化整为零，而不是一次性完成，这是很累而且很困难的事情。

2.任务一是我认为本次实验最难的一部分，因为他综合了我们之前学过的循环、选择（分支）、函数调用、嵌套等，是对我们能力的一个综合测验。

3.对于一个较复杂的程序来说，要进行多次的测试运行，我在任务一中失败了n次，测验了n次，每次一点一滴的改正，想到更好的解决方式，这是我在本次任务中学到的很重要的东西。

4.**不要轻易give up!**失败并不可怕。

# 任务二 采用递归和非递归的方式实现n！

## 2.1伪代码

1.已知用户输入n（正整数n>=1），总乘积为sum

2.求1\*2\*3\*…\*n的值

3.初始值n每次减一与前面的数相乘赋值给sum，直到n为1

4.输出总值sum

## 2.2程序的运行和测试

（1）我们先给出非递归算法，即运用循环来进行阶乘（for or while of do while）

（2）我这里采用了for循环来进行累乘求n的阶乘

（3）这里我给出了1 3以及-10的运行测试（以1，大于1，小于0的数为代表）

（4）相关代码在附录二中

**下面给出非递归算法**

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

下面给出递归算法

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

（5）由于负数测试同根同源是一样的，递归算法我没有给出负数 的错误处理。可以看出，我们的非递归算法并没有用到本节课学习到的知识，即调用函数。而递归算法则调用了自定义函数，并且大大简化了步骤。

## 2.3总结

1.任务二相对来说比较简单，其中非递归用到了我们之前学习的循环结构，可以说现在对于我们来说轻车熟路。

2.递归算法调用了自定义函数，起到了简化主函数的作用，并且要比循环结构更加方便。

3.要注意到自定义的函数要在main’之前给出定义，否则会报错。

# 任务三 做出三个数的排序函数sort（）

## 3.1伪代码

1.已知用户输入的三个数a，b，c（定义为double类型）

2.用户选择升序降序（即给出菜单选项）

3.通过if’条件句实现abc的赋值来进行排序

4.按照升序（降序）输出abc

## 3.2程序的运行和测试

（1）这里我给出了菜单选项，即让用户选择升序或者降序排列来让程序实用性更强。

（2）自定义了sort\_up && sort\_down两个自定义函数（利用赋值来实现）

（3）相关源代码在附录三中给出

下面给出运行结果

文本

描述已自动生成

文本

描述已自动生成

（4）这里给出了升序和降序的分别测试，即菜单选项和函数调用的准确性。

## 3.3总结

1.任务三的难度主要集中在如何判断三个数的大小并定义函数，其中方法有很多，我这里使用了不断赋值达到大小关系，可能每个人都有不同的方法，但只要解决了这个问题，整个任务便迎刃而解了

2.菜单选项和函数调用简直是绝配！放在一块使用让程序实用了很多。

3.另外，我在任务中犯了很多低级的错误，导致报错了很多次，像**scanf的使用中没有用&符号导致报错**，所以在程序中还是要仔细。低级错误要少犯。

# 任务四 猴子吃桃问题（自选任务）

## 4.1伪代码

1.已知第十天只剩下1个桃子

2.上一天的数量num=今天的数量2\*(n+1)

3.重复计算9次即可得到第一天采摘的数量num

4.输出num

## 4.2程序的运行和测试

（1）这里我采用了for循环，简单便捷，当然也可采用while等循环进行计算，或者将循环体做成函数调用也是一样的道理。但是程序比较简单就不必进行函数调用了。

（2）下面给出运行结果

（3）相关源代码在附录四中给出



本次任务较为简单，不再赘述

## 4.3总结

本次任务简单，但是要注意一个点，那就是到底循环几次的问题，是10次还是9次，我们考虑到第十天想吃的时候只剩下一个，而不是第十天吃完，其实是循环9次，这是需要注意的问题。

# 任务五 利用函数调用输出四个数的最大值（自选）

## 5.1伪代码

1.定义两个数的比较最大值函数max2

2.调用max2函数，并且定义四个数最大值函数max4

3.根据max2的比较，输出4个数中最大值m

4.在main函数中调用max4，输出m

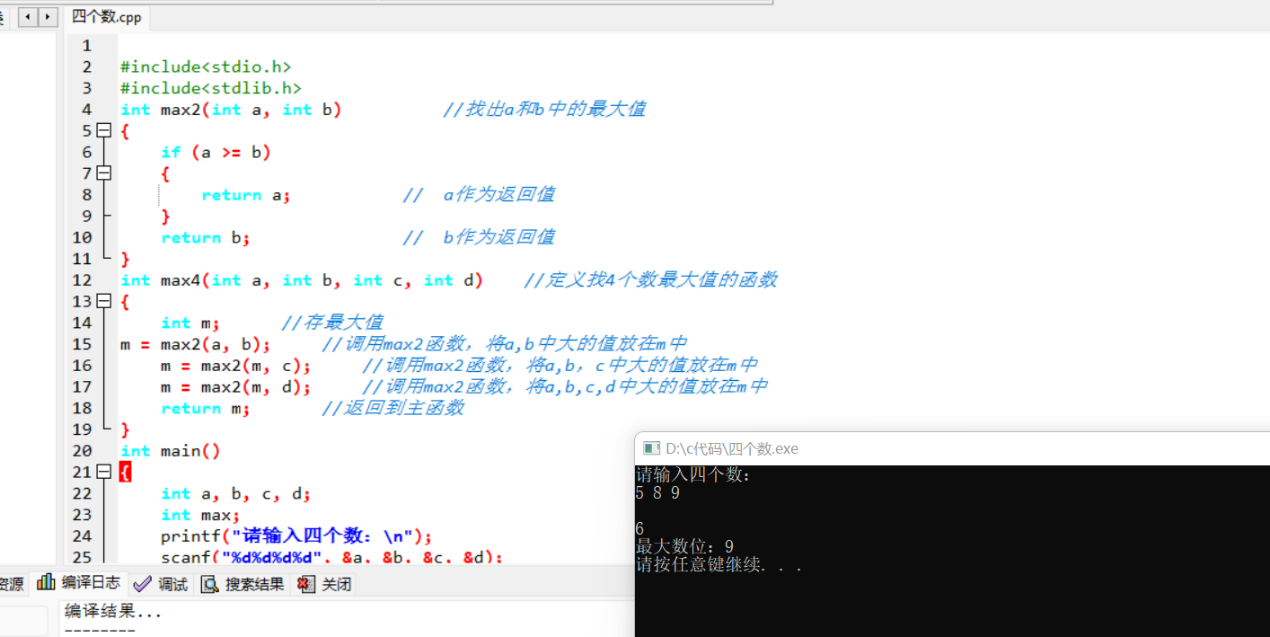
## 5.2程序的运行和测试

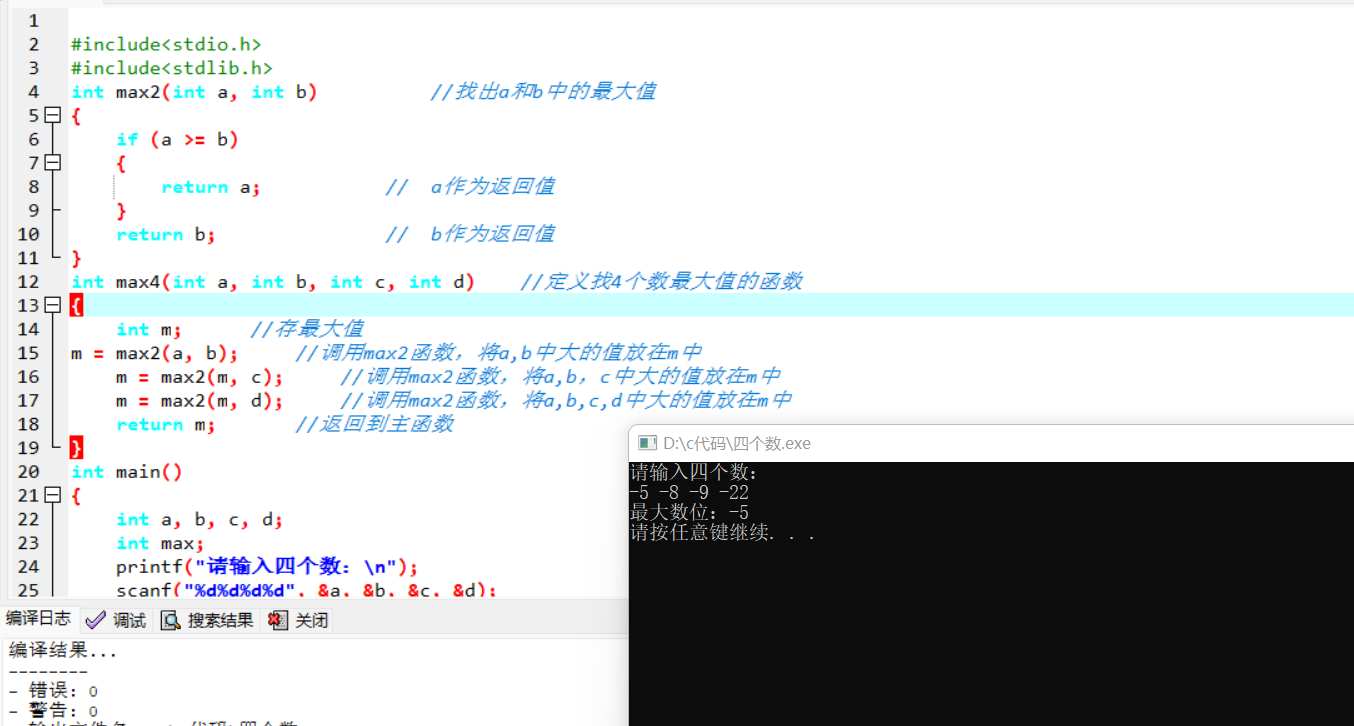
（1）这里我定义了两个函数max2和max4，并且在max4中套用了max2

（2）测试给出正数和负数的测试

（2）下面给出运行结果

（3）相关源代码在附录五中给出





## 5.3总结

1.函数可以套用，这是需要注意的一点

2.如果自定义函数在main上部，是可以在头部不声明的

3.注意函数格式（），{}，都不可以缺

4.如果给了函数返回值，那么一定要给出return语句！！

# 附录

## 附录一

#include <stdio.h>  
#include <stdlib.h>  
  
int judge(int year);  
int date\_num(int y,int m,int d);  //三个自定义函数   
int num\_date(int year,int sum);  
/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/  
  
      int main(int argc, char \*argv[]) {  
       int operation;  
      do{  
    
       system("cls");  
      int year,month,day,sum;  
        
      printf("请输入您的选项：\n");     //菜单选项   
  printf("1:日期转换成天数\n");  
  printf("2:天数转换成日期\n");   
  printf("0:退出\n");  
       fflush(stdin);  
       scanf("%d",&operation);  
       if(operation==1)  
   {  
    printf("请输入 年 月 日：(三个数字)\n");  
   scanf("%d%d%d",&year,&month,&day);  
   sum=date\_num(year,month,day);  
   printf ("天数为：%d",sum);  
   }  
      else  
      {  
       printf("请输入 年 天数(两个数字)\n");  
       fflush(stdin);  
  scanf("%d%d",&year,&sum);  
  day=num\_date(year,sum);  
  month=day/100,day%=100;  
  printf("日期为：%d月%d日",month,day);  
     
  }  
  getchar();  
      system("pause");  
  }while(operation!=0);  
return 0;  
}  
  
  
int judge(int year){                     //第一个自定义函数：判断是否闰年。   
if((year%4==0&&year%100!=0)||(year%400==0))  
return 1;  
else  
return 0;  
}  
  
  
  
int date\_num(int y,int m,int d)    {         //第二个自定义函数，即日期转换成天数。   
int sum=0;  
for(int i=1;i<m;i++)  
{  
if(i==4||i==6||i==9||i==11)  
sum+=30;  
else if(i==2)  
{  
  if(judge(y)==0)  
   sum+=28;  
   else  
   sum+=29;  
}  
        else  
           sum+=31;  
       }  
      sum+=d;  
        
      return sum;  
  }  
        
        
      int num\_date(int year,int sum)             //第三个自定义函数，即天数转换成日期   
      {  
         int i=1,j,t,d;  
         while(sum>0)  
         {  
          if(i==4||i==6||i==9||i==11)  
          {  
          sum-=30;  
          t=30;  
}  
          else if(i==2)  
          {  
          if(judge(year)==0)  
          {  
          sum-=28;  
          t=28;  
}  
          else  
          sum-=29;  
          t=29;  
}  
else{  
sum-=31;  
t=31;  
}  
i+=1;  
}  
                if(sum==0)  
   {  
    j=t;  
   }  
   else  
   {    
      j=sum+t;  
      i--;  
   }  
   d=i\*100+j;  
   return d;  
  }

## 附录二

#include <stdio.h>  
#include <stdlib.h>  
  
  
int main(int argc, char \*argv[]) {  
  
int n,sum=1;                              //定义n,sum为整型变量，且sum初值为1.   
printf("请输入一个正整数：");  
scanf("%d",&n);                           //取n的地址   
if(n>=1){                                  //判定条件，即n是否为正数   
  
for(n;n>=1;n--){                           //循环条件，即从n到1   
              
sum=sum\*n;                            //每次乘n-1的值    
}   
printf("%d",sum);                          //输出sum的值  
}  
           
else{                                      //判定为F 的执行语句   
printf("输入错误\n");  
}   
  
  
  
return 0;  
  
}

……………………………………………………………………………………………………………………………………………………………………………………………

 #include <stdio.h>  
#include <stdlib.h>  
int recursive(int n);  
   
 int main(int argc, char \*argv[]) {  
  int n;  
  printf("请输入一个正整数：");   
    
    scanf("%d",&n);  
      
     printf("%d",recursive(n));            //输出递归函数的值   
return 0;  
}  
  
int recursive(int n)                 //定义递归函数   
{  
if(n==1)                             //if条件句判定   
return 1;  
else                                 //n!=1时的运算   
return n\*recursive(n-1);  
  
}

## 附录三

#include <stdio.h>  
#include <stdlib.h>  
double a,b,c,n;  
int m;  
void sort\_down();  
void sort\_up();  
  
  
int main(int argc, char \*argv[]) {  
  
printf("1:升序\n");  
    printf("2:降序\n");  
printf("请输入数字选项：\n");  
scanf("%d",&m);  
if(m==1){  
printf("请输入三个数字：\n");  
scanf("%lf%lf%lf",&a,&b,&c);  
sort\_up();  
  
}  
else if (m==2){  
  
printf("请输入三个数字：\n");  
scanf("%lf%lf%lf",&a,&b,&c);  
sort\_down();  
  
}  
return 0;  
}  
  
  
void sort\_up(){  
  
if(a>b){  
n=b;  
b=a;  
a=n;  
}  
if(a>c){  
n=c;  
c=a;  
a=n;  
}  
if(b>c){  
n=c;  
c=b;  
b=n;   
}  
printf("%lf %lf %lf ",a,b,c);  
return;  
}  
  
void sort\_down(){  
  
if(a>b){  
n=b;  
b=a;  
a=n;  
}  
if(a>c){  
n=c;  
c=a;  
a=n;  
}  
if(b>c){  
n=c;  
c=b;  
b=n;   
}  
printf("%lf %lf %lf ",c,b,a);  
return;  
}

## 附录四

#include <stdio.h>  
#include <stdlib.h>  
  
int main(int argc, char \*argv[]) {  
int i,sum=1;  
for(i=1;i<=9;i++){         //采用for循环 进行九次重复运算   
  
sum=(sum+1)\*2;         //每天剩下的果子加上多吃的一个，再乘2即为前一天的果子书。   
  
}  
printf("第一天摘得果子数为：%d",sum);  //输出sum   
  
return 0;  
  
}

## 附录五

#include<stdio.h>  
#include<stdlib.h>  
int max2(int a, int b) //找出a和b中的最大值  
{  
if (a >= b)  
{  
return a; // a作为返回值  
}  
return b; // b作为返回值  
}  
int max4(int a, int b, int c, int d) //定义找4个数最大值的函数  
{  
int m; //存最大值m = max2(a, b); //调用max2函数，将a,b中大的值放在m中  
m = max2(m, c); //调用max2函数，将a,b，c中大的值放在m中  
m = max2(m, d); //调用max2函数，将a,b,c,d中大的值放在m中  
return m; //返回到主函数  
}  
int main()  
{  
int a, b, c, d;  
int max;  
printf("请输入四个数：\n");  
scanf("%d%d%d%d", &a, &b, &c, &d);  
max = max4(a, b, c, d); //调用max4函数  
printf("最大数位：%d\n", max);  
system("pause");  
return 0;