**实验三 控制结构-选择和循环的综合应用**

**姓名\_\_\_\_\_\_\_\_\_胡子豪\_\_\_\_\_\_\_\_\_\_学号\_\_\_302024315178\_\_\_\_\_\_**

**备注：将对应题的源代码放入题号对应的文件夹，并完成该文档后将文档重命名为：实验三+学号+姓名 上传到学习通**

**实验目的：**

**1）掌握选择控制结构if-else、switch-case的语法和基础应用；**

**2）掌握循环控制结构while、do-while、for的语法和基础应用；**

**3）掌握使用选择和循环控制的混合编程、嵌套使用，构造较复杂的问题解。**

* **Part One（50分）**

**3-1（10分）给定年份year，月份month，编程计算year年的month月有几天？**

1. 使用if-else结构实现
2. 使用switch-case结构实现

* **5分-完整程序解1）if-else：（贴代码）**
* #include<iostream>
* using namespace std;
* int main(){
* int y,m,d;
* cout<<"请输入年份和月份："<<endl;
* cin>>y>>m;
* if(y>=1 && m>=1 && m<=12){
* if(m==1||m==3||m==5||m==7||m==8||m==10||m==12) d=31;
* else if(m==4||m==6||m==9||m==11) d=30;
* else{
* if((y%4==0 && y%100!=0)||y%400==0) d=29;
* else d=28;
* }cout<<y<<"年"<<m<<"月有"<<d<<"天"<<endl;
* }
* else cout<<"输入的年月不合规"<<endl;
* return 0;
* }
* **5分-完整程序解2）switch-case：（贴代码）**
* #include<iostream>
* using namespace std;
* int main(){
* int y,m,d;
* cout<<"请输入年份和月份："<<endl;
* cin>>y>>m;
* if(y>1)
* {
* if((y%4==0 && y%100!=0)||y%400==0){
* switch (m)
* {
* case 1:
* case 3:
* case 5:
* case 7:
* case 8:
* case 10:
* case 12:d=31;break;
* case 4:
* case 6:
* case 9:
* case 11:d=30;break;
* case 2:d=29;break;
* default:cout<<"月份不合规"<<endl;
* break;
* }
* }
* else{
* switch (m)
* {
* case 1:
* case 3:
* case 5:
* case 7:
* case 8:
* case 10:
* case 12:d=31;break;
* case 4:
* case 6:
* case 9:
* case 11:d=30;break;
* case 2:d=28;break;
* default:cout<<"月份不合规"<<endl;
* break;
* }
* }
* cout<<y<<"年"<<m<<"月有"<<d<<"天"<<endl;
* }
* else cout<<"年份不合规"<<endl;
* return 0;
* }

**3-2 （20分）回顾课堂程序s=1+2+3+…+n 的程序解决模型**(n为正整数)**，请在课堂学习的基础上实现:**

1）连续的乘可以设计为与累加相似的计算模型，即累乘的不变式可构造为s=s\*▲,请编写程序完成s= n! = 1\*2\*3\*…\*n的计算(n≥1)。（使用while和do-while两种方案）

2）编写程序完成 S=（n≥1）（使用while和for两种方案，**请预习for的语法格式，或者等周三课程完成后完成for实现**）

3）编写程序完成 S= (求解到 <0.001234 为止)**（while，do-while，for三种方案任选一种实现）**

**思考：总结2）3）这种累加分数的问题中应注意的问题；总结给出累加项数和累加通项的终止值 控制循环的差别。**

* **3分-程序解1）while：**
* #include<iostream>
* using namespace std;
* int main(){
* int i=1,n;
* unsigned long sum=1;
* cout<<"请输入累乘的次数n"<<endl;
* cin>>n;
* while (i<=n)
* {
* sum=sum\*i;
* i=i+1;
* }
* cout<<sum<<endl;
* }
* **3分-程序解1）do-while：**
* #include<iostream>
* using namespace std;
* int main(){
* int i=1,n;
* unsigned long sum=1;
* cout<<"请输入累乘的次数n"<<endl;
* cin>>n;
* do{
* sum=sum\*i;
* i=1+i;
* }
* while (i<=n);
* cout<<sum<<endl;
* return 0;
* }
* **3分-程序解2）while：**
* #include<iostream>
* using namespace std;
* int main(){
* int n;
* double i=1,sum=0;
* cout<<"请输入次数n"<<endl;
* cin>>n;
* while(i<=n)
* {
* sum=sum+1/i;
* i=i+1;
* }
* cout<<sum<<endl;
* }
* **3分-程序解2）for：**
* #include<iostream>
* using namespace std;
* int main(){
* int n;
* double i=1,sum=0;
* cout<<"请输入计算的次数n"<<endl;
* cin>>n;
* for(i=1;i<=n;i++)
* sum=sum+1/i;
* cout<<sum<<endl;
* return 0;
* }
* **4分-程序解3）：**
* #include<iostream>
* using namespace std;
* int main(){
* double i=1,sum=0;
* while(i<(1/0.001234))
* {
* sum=sum+1/i;
* i=i+1;
* }
* cout<<sum<<endl;
* }
* **4分-思考总结回答：**

**应注意分母不为零，储存的格式应该用double**

**差别在于条件判断的部分根据条件不同用不同的判断方式**

**3-3、(10分)如果一个3位正整数等于其各个数字的3次方的和，则称这个数为阿姆斯特朗数(也称自恋型数),也称为水仙花数，如407=43+03+73。请编写程序求100~999以内的所有阿姆斯特朗数，要求输出数据之间使用逗号（，）分隔，末尾没有逗号。**

* **Tips1:While、do-while、for均可以利用循环控制给出待处理的数据集。比如1+2+3+…+100的累加就给出了需要处理的1，2，3…100数据集。选择一种循环控制实现水仙花数的求解，并思考对于这种数据集为连续整数的处理，哪种循环结构更直观、可读性更好？**
* **Tips2：题3-3需要一个分离数位的技巧，考虑如何将一个数num的各个数字分离出来？我们可以利用它们所对应的位权来处理。如分离个位，那么就是num%10; 分离十位，那么就是num/10%10; 分离百位，那么就是num/100%10，依次类推。请你思考，找到分离数位相关程序的求解策略。**

**拓展知识：水仙花数只是自幂数（n位的正整数是其各位数字的n次方和）的一种。其他位数的自幂数叫法如下：**

**一位自幂数：独身数；**

**三位自幂数：水仙花数；**

**四位自幂数：四叶玫瑰数；**

**五位自幂数：五角星数；**

**六位自幂数：六合数；**

**七位自幂数：北斗七星数；**

**八位自幂数：八仙数；**

**九位自幂数：九九重阳数；**

**十位自幂数：十全十美数。**

**其他的自幂数请自行探索。**

* **程序解：**
* #include<iostream>
* using namespace std;
* int tri(int x){
* int y;
* y=x\*x\*x;
* return y;
* }
* int main(){
* int i=100;
* bool flag=1;
* for(i=100;i<=999;i++)
* {if((tri(i/100)+tri(i/10%10)+tri(i%10))==i)
* {if(!flag) cout<<",";
* cout<<i;
* flag=0;
* }
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

**文本

描述已自动生成**

**3-4、(10分)猴子爬杆**

|  |
| --- |
| 基本描述 |
| 有一只特别能爬杆的小猴子。整天梦想能借着杆子爬到天上去。它给自己列了个计划：从地面开始，第一天爬5米；第二天爬前一天爬的2倍高多1米；第三天爬前一天爬的2倍高少2米；第四天又爬前一天的2倍高多3米；….，就这样一日复一日。你帮小猴子写个程序算一算它每天爬多少米？已经离地面有多高了吗？ |
| 输入描述 |
| 多值输入，每个值描述某天是小猴子爬杆的第几天？ (1≤第几天的值≤29) |
| 输出描述 |
| 对应每个输入值，输出小猴子当天爬了几米，已经离地面有多少高了？  每行的具体格式请参考样本输出描述。每行行末回车。  所有数值均不超过unsigned int |
| 样本输入 |
| 1  3  14 |
| 样本输出 |
| 5,5  20,36  42785,85558 |

* **程序解：**
* #include<iostream>
* using namespace std;
* int pdgd(unsigned x){
* unsigned i,t=5;
* for(i=2;i<=x;i++)
* {
* if(i%2==0) t=t\*2+(i-1);
* else t=t\*2-(i-1);
* }
* return t;
* }
* int zgd(unsigned x){
* unsigned i,t=5,sum=5;
* for(i=2;i<=x;i++)
* {
* if(i%2==0) t=t\*2+(i-1);
* else t=t\*2-(i-1);
* sum=sum+t;
* }
* return sum;
* }
* int main(){
* unsigned a;
* while(cin>>a)
* cout<<pdgd(a)<<","<<zgd(a)<<endl;
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

**文本

描述已自动生成**

* **Part Two（50分）**

**3-5、(10分)丢掉质数**

|  |
| --- |
| **基本描述** |
| 小A接到了一个任务，需要处理一个N规模的连续整数集合，要把其中绝对值为质数的数据丢掉。 |
| **输入描述** |
| 输入多组数据，每组包括数据集规模N，数据集起始数据X。即表示要处理的原始数据集包含从X开始的N个整数。 |
| **输出描述** |
| 对于每个数据集，给出丢掉质数后的数据列表，使用{}和，表示处理后的数据集。行末回车。 |
| **样本输入** |
| 8 3  5 -22  7 -3 |
| **样本输出** |
| {4,6,8,9,10}  {-22,-21,-20,-18}  {-1,0,1} |

* **程序解：**
* #include<iostream>
* #include<cmath>
* using namespace std;
* int pdzs(int x){
* int y,i;
* bool flag=false;
* if(x<0) x=abs(x);
* if(x==0||x==1) flag=true;
* y=int(sqrt(x));
* for(i=2;i<=y;i++)
* if(x%i==0) flag=true;
* return flag;
* }
* int main(){
* cout<<"请输入数据规模和起始数据："<<endl;
* int a,b,c;
* bool f=false;
* while(cin>>a>>b)
* {
* f=false;
* cout<<"{";
* c=a+b;
* for(b;b<c;b++)
* {
* if(pdzs(b)){
* if(f) cout<<',';
* cout<<b;
* f=true;
* }
* }
* cout<<"}"<<endl;
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

**文本

描述已自动生成**

**3-6、（10分） 计算s=**

**其中x的值由用户输入。设计不同的程序完成求解1）求当n=10的s值；2）求当最后项的精度小于10-5的s值。**

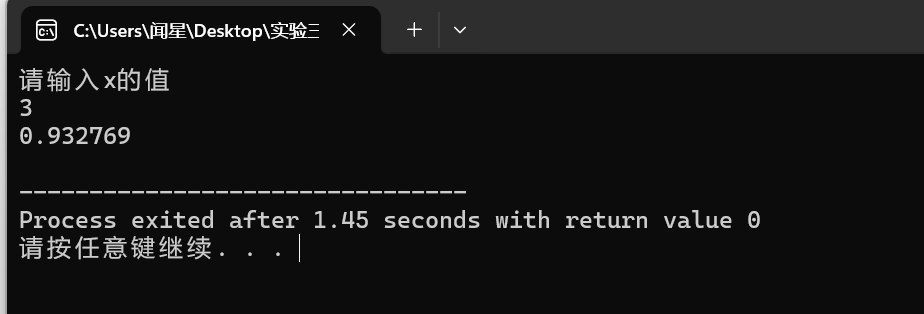
**注意通过比较前后项差异去寻找计算△的方法。**

* **5分-程序解：1）**
* #include<iostream>
* #include<cmath>
* using namespace std;
* double jc(double x){
* int i=1,sum=1;
* while (i<=x)
* {
* sum=sum\*i;
* i=i+1;
* }
* return sum;
* }
* int main(){
* int n=10,i;
* double s,x;
* cout<<"请输入x的值"<<endl;
* cin>>x;
* for(i=1;i<=10;i++)
* s=(pow((-1),i+1)\*(pow(x,i)/jc(i)))+s;
* cout<<s<<endl;
* }
* **运行截图：**

**图形用户界面, 文本, 应用程序

描述已自动生成**

* **5分-程序解：2）**
* #include<iostream>
* #include<cmath>
* using namespace std;
* long long jc(int x){
* int i=1,sum=1;
* while (i<=x)
* {
* sum=sum\*i;
* i=i+1;
* }
* return sum;
* }
* double element(double x,double n){
* double y;
* y=pow(x,n)/jc(n);
* return y;
* }
* int main(){
* int n=1,flag=1;
* double x,s=0;
* cout<<"请输入x的值"<<endl;
* cin>>x;
* for(n=1;element(x,n)>=0.00001;n++)
* {
* s=s+flag\*element(x,n);
* flag=flag\*(-1);
* }
* cout<<s<<endl;
* }
* **运行截图：**

****

**3-7、(10分)如果一个正整数各个数字位和是3的倍数，那么这个正整数就可以被3整除。构造完整程序判断输入数据num是否为3的倍数。程序运行一次可以进行多次判定，直到输入Ctrl+Z。**

**连续测试多组数据：**

**输入：5**

**输出：不是3的倍数**

**输入：2345**

**输出：不是3的倍数**

**输入：12345**

**输出：是3的倍数**

**输入：Ctrl+Z结束程序**

* **程序解：**
* #include<iostream>
* #include<string>
* using namespace std;
* int main(){
* string a;
* int i,sum=0;
* cout<<"请输入一个正整数："<<endl;
* while(cin>>a){
* sum=0;
* for(i=a.size()-1;i>=0;i--)
* {
* sum=sum+a[i]-'0';
* }
* if(sum%3==0)cout<<"是3的倍数"<<endl;
* else cout<<"不是3的倍数"<<endl;
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

**文本

描述已自动生成**

**3-8、(10分)叠三角形**

|  |
| --- |
| 基本描述 |
| 某个建筑队想要建造一个特别的建筑，由两个等腰三角形堆叠而成。现在题目希望有个程序可以把他们想要的建筑形状画出来！现在请你帮他们来做这件事。 |
| 输入描述 |
| 多组数据输入，每组数据输入底座三角形的高度m，上层三角形的高度n。其中，n,m均为正整数且满足n≤m≤50。 |
| 输出描述 |
| 对于每组数据，画出对应的叠加的等腰三角形。上下两个三角形的构成规律相同，由字符’\*’为材料建造，三角形居中对齐，每个堆叠工作后跟一个换行。具体参看样本示范。 |
| 样本输入 |
| 3 2  5 3 |
| 样本输出 |
| \*  \*\*\*  \*  \*\*\*  \*\*\*\*\*  \*  \*\*\*  \*\*\*\*\*  \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\* |

* **程序解：**
* #include<iostream>
* #include<iomanip>
* using namespace std;
* int main(){
* int m,n,i,j,k;
* cout<<"请输入下上两个三角形的高："<<endl;
* while(cin>>m>>n)
* {
* if(n>m || m>=50) cout<<"输入高度不合理，请重新输入"<<endl;continue;
* for(i=1;i<=n;i++)
* {
* for(j=1;j<=m-i;j++)
* {
* cout<<" ";
* }
* for(k=1;k<=(2\*i-1);k++)
* {
* cout<<"\*";
* }
* cout<<endl;
* }
* for(i=1;i<=m;i++)
* {
* for(j=1;j<=m-i;j++)
* {
* cout<<" ";
* }
* for(k=1;k<=(2\*i-1);k++)
* {
* cout<<"\*";
* }
* cout<<endl;
* }
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

**图形用户界面, 文本

描述已自动生成**

**★3-9、（10分）万年历 ：由用户输入一个日期（包含y年m月d日信息），请编写一个万年历程序能计算出这一天是星期几？**

* **编程提示：假设用户输入的日期为y年m月d日。我们以公元的元年元月元日（1年1月1日）为第一天，那天是星期一。星期以7为周期。所以，我们考虑y年m月d日有几天，求得的总天数以7为周期取余数，即可获得对应的星期几。**

**计算天数由三部分构成：**

**（1）y-1年包含的天数。 days1=（y-1）\* 365+闰年的数量；因为闰年有366天。**

**（2）m-1月包含的天数。 days2=1月天数+2月天数+…+m-1月天数。**

**（3）days3=d天。**

**因此， 总的天数＝days1+day2+days3，然后将求得的总天数进行模7运算，即得以７为周期的余数。得到的值与星期几之间有如下对应关系： 0－星期日；1－星期一；2－星期二；3－星期三；4－星期四；5－星期五；6－星期六。**

* **程序对应的流程图如下：**

真

输入ｙ,ｍ,ｄ值（年，月，日）

ｄａｙｓ１＝(ｙ－１)＊３６５＋闰年的数量

ｄａｙｓ２初始化为０，ｔ＝１

ｔ<＝ｍ－１

ｄａｙｓ２＝ｄａｙｓ２＋ｔ月的天数

ｔ＝ｔ＋１

ｄａｙｓ＝ｄａｙｓ１＋ｄａｙｓ２＋ｄ

求ｄａｙｓ对７的余数

假价

**开始**

**结束**

将余数转换成对应的星期并输出

**整个流程图中包含一个循环执行的片断，请找到整个片断。**

**我们可以使用**

循环判断条件

需要重复执行的语句序列

真

假

**while（循环判断条件）{**

**需要重复执行的语句序列**

**}**

**这样的结构来解决需要重复执行某些语句。请尝试着用一下。**

* **其中:**

**1）闰年的数量=包含4倍数的量-包含100倍数的量+包含400倍数的量**

**2）t月的天数K由第3-1题程序的求解获得，抽取出有效的程序片段，用在t月天数的计算中。**

* **思考：**

**1）如何求days对7的余数？**

**2）如何将余数转换为对应的星期并输出？**

**3）如何保证数据输入合法？（y,m,d必须为非负整数）**

**★4）如果days值很大，以至于系统的long也无法处理，该怎么办？**

★★5）考虑如何输出一个月的日历？即给出ｙ年ｍ月，将ｍ月日历都显示出来。如 2024年10月，显示格式为：

**一 二 三 四 五 六 日**

**1 2 3 4 5 6**

**7 8 9 10 11 12 13**

**14 15 16 17 18 19 20**

**21 22 23 24 25 26 27**

**28 29 30 31**

**输出格式控制提示：**

**方法一，可以使用输出适当的空格来完成输出的排版；方法二，在输出流中可以使用setw控制输出宽度，setw用法参考课本或者百度。**

**★★★6）考虑如何输出一年的年历？即给出y年，将该年的所有月都输出。**

* **程序解：**
* #include<iostream>
* #include<iomanip>
* using namespace std;
* int rngs(int x){//计算闰年的数量
* int i,y=0;
* for(i=1;i<=x-1;i++)
* {
* if((i%4==0 && i%100!=0)|| i%400==0) y=y+1;
* }
* return y;
* }
* int sumM(int y,int m){//计算输入年份的前几个月的天数
* int a,i,days2=0;
* for(i=1;i<=m-1;i++){
* switch(i){
* case 1:case 3:case 5:case 7: case 8:case 10:case 12:days2=days2+31;break;
* case 4:case 6:case 9:case 11:days2=days2+30;break;
* case 2:
* {
* if((y%4==0 && y%100!=0)|| y%400==0)days2=days2+29;
* else days2=days2+28;
* }
* }
* }
* return days2;
* }
* int main(){
* long long day,day1,day2;
* int y,m,d;
* cout<<"请输入年月日:"<<endl;
* do{cin>>y>>m>>d;}while(y<1 || m<1 || d<=1 );
* day1=365\*(y-1)+rngs(y);
* day2=sumM(y,m);
* day=day1+day2+d;
* switch (day%7)
* {
* case 0:cout<<"星期日"<<endl;break;
* case 1:cout<<"星期一"<<endl;break;
* case 2:cout<<"星期二"<<endl;break;
* case 3:cout<<"星期三"<<endl;break;
* case 4:cout<<"星期四"<<endl;break;
* case 5:cout<<"星期五"<<endl;break;
* case 6:cout<<"星期六"<<endl;break;
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

图形用户界面, 文本

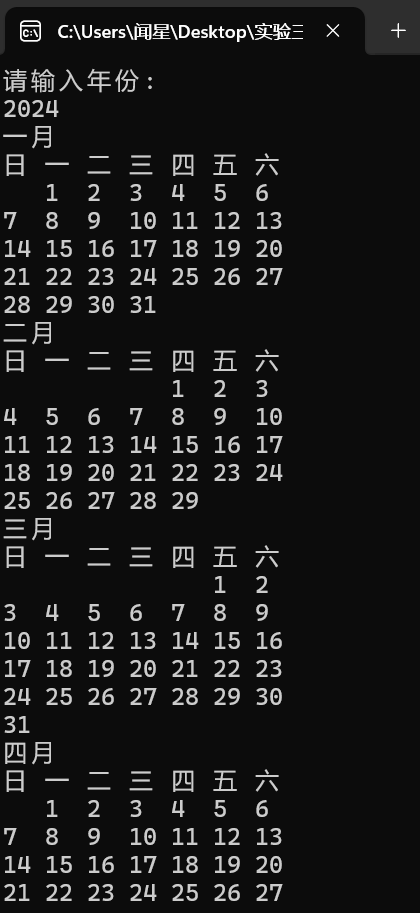
描述已自动生成

（5）输出一个月的万年历

* **程序解：**
* #include<iostream>
* #include<iomanip>
* using namespace std;
* int rngs(int x){//计算闰年的数量
* int i,y=0;
* for(i=1;i<=x-1;i++)
* {
* if((i%4==0 && i%100!=0)|| i%400==0) y=y+1;
* }
* return y;
* }
* int sumM(int y,int m){//计算输入年份的前几个月的天数
* int a,i,days2=0;
* for(i=1;i<=m-1;i++){
* switch(i){
* case 1:case 3:case 5:case 7: case 8:case 10:case 12:days2=days2+31;break;
* case 4:case 6:case 9:case 11:days2=days2+30;break;
* case 2:
* {
* if((y%4==0 && y%100!=0)|| y%400==0)days2=days2+29;
* else days2=days2+28;
* }
* }
* }
* return days2;
* }
* int week(int y,int m){//计算这个月的第一天是周几
* long long day,day1,day2;
* int week,d=1;
* day1=365\*(y-1)+rngs(y);
* day2=sumM(y,m);
* day=day1+day2+d;
* week=day%7;
* return week;
* }
* int main(){
* int y,m,d,md;
* const char\* weekname[]={"日", "一", "二", "三", "四", "五", "六"};
* cin>>y>>m;
* d=week(y,m);
* switch(m){
* case 1:case 3:case 5:case 7: case 8:case 10:case 12:md=md+31;break;
* case 4:case 6:case 9:case 11:md=md+30;break;
* case 2:
* {
* if((y%4==0 && y%100!=0)|| y%400==0)md=md+29;
* else md=md+28;
* }
* }
* for(int i=0;i<=6;i++)
* cout<<weekname[i]<<" ";
* cout<<endl;
* for(int i=0;i<d;i++)
* {
* cout<<setw(3)<<" ";
* }
* for(int i=1;i<=md;i++)
* {
* cout<<setw(3)<<setiosflags(ios::left)<<i;
* if((i+d)%7==0)cout<<endl;
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

（6）输出一年的万年历

* **程序解：**
* #include<iostream>
* #include<iomanip>
* using namespace std;
* int rngs(int x){//计算闰年的数量
* int i,y=0;
* for(i=1;i<=x-1;i++)
* {
* if((i%4==0 && i%100!=0)|| i%400==0) y=y+1;
* }
* return y;
* }
* int sumM(int y,int m){//计算输入年份的前几个月的天数
* int a,i,days2=0;
* for(i=1;i<=m-1;i++){
* switch(i){
* case 1:case 3:case 5:case 7: case 8:case 10:case 12:days2=days2+31;break;
* case 4:case 6:case 9:case 11:days2=days2+30;break;
* case 2:
* {
* if((y%4==0 && y%100!=0)|| y%400==0)days2=days2+29;
* else days2=days2+28;
* }
* }
* }
* return days2;
* }
* int week(int y,int m){//计算这个月的第一天是周几
* long long day,day1,day2;
* int week,d=1;
* day1=365\*(y-1)+rngs(y);
* day2=sumM(y,m);
* day=day1+day2+d;
* week=day%7;
* return week;
* }
* int main(){
* int y,m,d,md;
* const char\* weekname[]={"日", "一", "二", "三", "四", "五", "六"};
* const char\* monthname[]={"一月","二月","三月","四月","五月","六月","七月","八月","九月","十月","十一月","十二月"};
* cout<<"请输入年份:"<<endl;
* cin>>y;
* for(m=1;m<=12;m++){
* md=0;
* d=week(y,m);
* switch(m){
* case 1:case 3:case 5:case 7: case 8:case 10:case 12:md=md+31;break;
* case 4:case 6:case 9:case 11:md=md+30;break;
* case 2:
* {
* if((y%4==0 && y%100!=0)|| y%400==0)md=md+29;
* else md=md+28;
* }
* }
* cout<<monthname[m-1]<<endl;
* for(int i=0;i<=6;i++)
* cout<<weekname[i]<<" ";
* cout<<endl;
* for(int i=0;i<d;i++)
* {
* cout<<setw(3)<<" ";
* }
* for(int i=1;i<=md;i++)
* {
* cout<<setw(3)<<setiosflags(ios::left)<<i;
* if((i+d)%7==0)cout<<endl;
* }
* cout<<endl;
* }
* return 0;
* }
* **程序测试结果（运行结果截屏）：**

图片包含 日历

描述已自动生成