编程题

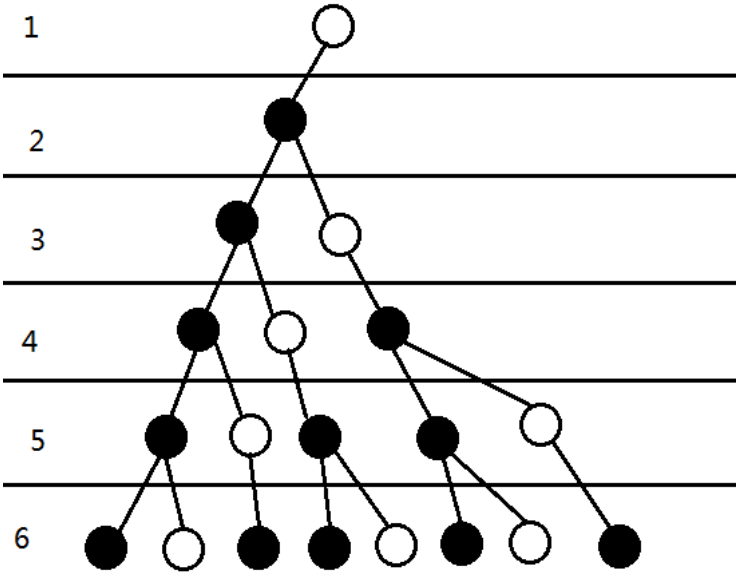
1，声明一个教师（Teacher）类和一个学生（Student）类，用多重继承的方式声明一个研究生（Graduate）派生类。教师类中包括数据成员name（姓名），age（年龄），title（职称）。学生类中包括数据成员name（姓名），age（年龄），score（成绩）。在定义派生类对象时给出初始化的数据（自己定），然后输出这些数据。初值自拟。

```
#include <iostream>
#include <string>
using namespace std;
class Teacher{
    public: Teacher(string nam,int a,string t)
    {
        name=nam;
        age=a;
        title=t;

    }
   void display()
   {
       cout<<"name:"<<name<<endl;
       cout<<"age"<<age<<endl;
      cout<<"title:"<<title<<endl;
    }
  protected:  string name;     int age;
  string title;

};
class Student{
    public:  Student(string nam,char s,float sco)
    {
        name1=nam;
        sex=s;
        score=sco;
    }
    void display1()
    {
        cout<<"name:"<<name1<<endl;
        cout<<"sex:"<<sex<<endl;
        cout<<"score:"<<score<<endl;

    }
    protected: string name1;    char sex;
    float score;
};


class Graduate:public Teacher,public Student {
    public: Graduate(string nam,int a,char s,string t,float sco,float w):Teacher(nam,a,t),Student(nam,s,sco),wage(w) {}
    void show( )
    {
        cout<<"name:"<<name<<endl;
        cout<<"age:"<<age<<endl;
        cout<<"sex:"<<sex<<endl;
        cout<<"score:"<<score<<endl;
        cout<<"title:"<<title<<endl;
        cout<<"wages:"<<wage<<endl;
```

```
    }
  private:
    float wage;
};
int main( )
{
    Graduate grad1("Wang-li",24,'f',"assistant",89.5,1234.5);
    grad1.show( );
    return 0;

}
```

2.古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

解析：
问题是这样的：古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一 对兔子， 假如
这个问题相信大家已经不在陌生了。很多博客里都有各种不同的解答方法。
最多的方法就是先列出最初几个月的兔子对数（注意是对数，不是个数）。如下所示：
1
1,1,2,3,5,8,13,21,34....
然后观察数据的规律，从而得出这样一个结论：从第三个月开始，兔子对数等于前面两个月的兔子对数之和。看到这里相信你已经有写
我要介绍的是，不知道这个规律的前提进行编程。
思路是这样的：
①、有一个笼子，我们逐一取出笼子中的一对兔子（当然这两只兔子的年龄是相同的）。
②、若它们的年龄大于或等于三个月，则生出一对小兔子。
③、将这两对兔子放入笼中。
④、所有兔子的年龄加1（上面出生的兔子年龄不加）。
⑤、5查看此时笼子中的兔子数量。



正确的算法如下图所示（建议最好画下来）：其中最左边的数表示月份，白色的圈表示未成熟的兔子，黑色的圆表示已经成熟，可以生育的兔子。

其中我们可以发现，每月已经成熟的兔子数量等于上个月的兔子数量，这是因为一对兔子过了一个月就会成熟，而已经成熟的兔子过了一个月还是成熟的。这样，因为每一对成熟的兔子在下一个月都会生出一对兔子，所以每月兔子增加的数量就是两个月前兔子的数量，这样把每个月兔子的数量排成一个数列，正好是著名的"斐波那契数列"。

```
int rabit(int n)
{
    if(n==1||n==2)
    {
        return 1;
    }
    else if(n>=3)
    {
         return (rabit(n-1))+(rabit(n-2));
    }
    return 0;
}
void main()
{
    for(int i=1;i<=12;i++)
    {
        printf("第"+i+"个月兔子对数为:"+rabit(i));
    }
}
```

3.有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。

```
#include<stdio.h>
main()
{
    int a[100];
    int i,n,p=0,q;
    printf("input number:");
    scanf("%d",&n);
    q=n;
    for(i=0;i<n;i++) a[i]=i+1;
    for(i=0;;i++)
    {
        if(i==n) i=0; //当i++一直到n时，肯定有一些没有被选到，比如我们输入8,第一轮是3,6被赋值0，当i=8时，继续下一轮//
        if(a[i]!=0) p++;//我们下面定义的是当循环到三时，就赋值0，所以这边等0的不考虑在内//
        else continue;
        if(p%3==0)//这个就是从0一直加，到三的倍数就赋值为0，从而就达到我们的目的//
         {a[i]=0;q--;} //上面q=n;表明q==n,只有一个为0就减一，为下面做铺垫//
        if(q==1) break; //当剩下最后一个就输出//
    }
    for(i=0;i<n;i++)
       if(a[i]!=0)
       printf("spare: %d\n\n",a[i]);
}
```

```
方法2：
#include<iostream>

using namespace std;

int main()
```

```
{
    int num[50];

    int i,j,k,m,n;

    int *p;

    cout<<endl<<"请输入总人数：<<endl;

    cin>>n;

    p=num;

    for(i=0;i<n;i++)
    {
        *(p+i)=i+1;   //以1至n为序，给每个人编号
    }

    i=0;    //i为每次循环时计数变量

    k=0;    //k为按1 2 3报数时的计数变量

    m=0;    //m为退出人数

    while(m<n-1)  //当退出人数比n-1少时（即未退出人数大于1时）执行循环体
    {
        if(*(p+i)!=0)
        {
            k++;
        }

        if(k==3)    //将退出人的编号置为0
        {
            *(p+i)=0;

            k=0;

            m++;
        }

        i++;

        if(i==n)
        {
            i=0;//报数到尾后i恢复为0
        }
    }

    while(*p==0)
    {
        p++;
    }

    cout<<"最后一个是"<<*p<<" 号！"<<endl;

    return 0;
}
```

方法3：

```
#include <stdio.h>
```

```
 int M = 3;

int main()
{
    int n, s = 0;
    scanf("%d", &n);
    for (int i = 2; i <= n; ++i)
        s = (s+M)%i;
    printf("%d\n", s+1);
    return 0;
}
```