1 if-else

```
boolean isLeapYear(int year){
    if(year % 400 == 0){
        return true;
    }
    else if(year %4 == 0 &&  year % 100 ! = 0){
            return true;
    }
    else{
        return false;
            }

        }
    }
}
```

switch-case不是if-else的语法糖，if-else底层实现是靠条件判断，编译器会生成代码顺序检查每个条件。而switch-case底层实现靠跳转，直接跳转到对应的case条件。

2 for-while

```
void print(int n){
        for (int i = 1; i <=n ; i++) {
            int Before;
            int Middle;
            if(i<=(n+1)/2){
                 Before=(n+1)/2-i;
                 Middle=(i-1)*2-1;
            }
            else{
                Before=i-(n+1)/2;
                Middle=(n-i)*2-1;
            }
            for (int j = 0; j < Before; j++) {
                System.out.print(" ");
            }
            System.out.print("*");
            if(i!=1&&i!=n){
                for (int z= 0; z <Middle ; z++) {
                    System.out.print(" ");
                }
                System.out.print("*");
            }
            System.out.println();
        }
    }
```

3递归和迭代

```
int  Fibonacci(int n){
        if(n==0){
            return 0;
        }
        else if(n==1){
            return 1;
        }
        else{
            return Fibonacci(n-1)+Fibonacci(n-2);
        }
    }
}
```

```
int Fibonacci(int n){
    if(n==0){
        return 0;
    }
    else if(n==1||n==2){
        return 1;
    }
    else{
        for (int i = 3; i <=n; i++) {
            int a=1,b=1,c;
            c=a+b;
            a=b;
            b=c;
            return c;
        }
    }
}
```

递归是函数调用自身，迭代是用循环结构。迭代内存效率高，只是重复执行代码，无需占用额外栈空间，而递归占用栈空间较大，容易发生栈溢出。

4 汉诺塔

```
void hanoi(int n){
      move(3,'A','C','B');//A做原柱，C做目标柱，B做辅助柱
    }
    void move(int n,char from,char to,char help){
        if(n==1){
            System.out.println(from+"->"+to);//如果n=1，直接从A柱移到C柱上
        }else {
            move(n - 1, from, help, to);//递归，A做原柱，B做目标柱，C做辅助柱，将n-1个铁饼移到B柱上。
            System.out.println(from + "->" + to);//将最大的也就是第n块铁饼移到C柱上，此时最大的铁饼就可以忽略，相当于有n-1块铁饼的情况
            move(n - 1, help, to, from);//再次递归，B做原柱，C做目标柱，A做辅助柱，将n-1块铁饼移到C柱上
        }
    }
}
```