

# 1. 运算符

1. 设计一个程序，输入三位数 $a$ ，分别输出百,十,个位。  
( $0 < a < 1000$ )

样例输入：

251

样例输出：

2

5

1

2. 设计一个程序，输入整数 $l$ ，求边长为 $l$ 的正方形面积，比直径为 $l$ 的圆形面积大多少。  
( $0 < l < 1000$ ， $\pi$ 取3.14，输出结果保留两位小数)

样例输入：

3

样例输出：

1.93

# 2. 判断

1. 设计一个程序，输入 $a, b, c$ 三个整数，输出最大的数。  
( $\text{MIN\_INT} < a, b, c < \text{MAX\_INT}$ )

样例输入：

1 3 2

样例输出：

3

2. 打分系统

在某次考试时，学生的提交成绩惨不忍睹

云海学长为了让大家及格可谓煞费苦心，他苦思三天三夜，终于想出了一套打分方案：

正确数量在0~10时，每题6分

正确数量在11~20时，第11~20题，每题2分

正确数量在21~40时，第21~40题，每题1分

但因为需要计分的作业太多了，希望你能帮帮云海学长，实现该计分程序。

输入做对的题目数量 $n$ ，输出得分。

( $0 \leq n \leq 40$ )

样例输入：

23

样例输出：

83

3. 东东哥上班

东东哥在上班途中，可以选择走路和骑车两种方式  
但他不清楚哪种方式比较快，因为骑车需要找车，开锁，停车 需要耽搁很多时间。  
设找到自行车，开锁，骑自行车需要花27秒，停车需要花23秒  
步行每秒1.2米，骑车每秒3.0米  
给出东东哥距离公司的距离，请问是骑车快还是走路快。

输入一个整数 $n$ ，表示到公司的距离  
如果骑车快，输出"骑车"  
如果走路快，输出"走路"  
如果一样快，输出"一样快"

样例输入：  
60  
样例输出：  
走路

### 3. 简单循环

#### 1. 求平均年龄

01星球有学长若干名，给出每个学长的年龄，求01星球学长的平均年龄，保留小数点后两位  
输入：  
第一行：整数 $n(1 < n < 100)$ ，表示人数  
之后 $n$ 行：每行一个整数 $a(15 < a < 55)$ ，表示第 $n$ 个学长的年龄  
输出：  
一个浮点数，保留两位小数。表示01星球学长平均年龄

样例输入：  
3  
18  
18  
17  
样例输出：  
17.67

2. 张三上课时走神，在作业上写了很多2222222222222222 结果被老师发现了  
张三连忙掩饰说自己在算 $2+22+222+2222+22222$ 的和，  
老师很欣慰，问他算出来没有  
张三很困扰，因为草稿纸写满了2 已经不够用了

大家快帮帮张三吧  
设计一个程序，输入 $n$ 和 $a$  求 $S_n$   
 $S_n = a + aa + aaa + \dots + \underbrace{aaaaa}_{n\text{个}a}$

输入：两个整数 $n, a$   
( $1 < n < 9$ ), ( $1 \leq a \leq 9$ )  
输出：  
 $S_n$ 的值

样例输入：  
5 2  
样例输出：  
24690  
样例解析： $S_n = 2 + 22 + 222 + 2222 + 22222 = 24690$

样例输入：

4 5

样例输出：

6170

样例解析： $S_n = 5 + 55 + 555 + 5555$

3. **super**家养了一对刚出生的兔子，兔子出生第3月开始每月都会生一对小兔子，小兔子出生后同样第3月开始也会每月生一对兔子

**super**想知道 如果兔子不死  $n$  月后家里会有多少对兔子

设计一个程序：输入 $n$ ，输出兔子数量

( $2 < n < 30$ )

样例输入：7

样例输出：13

样例输入：12

样例输出：144

4. 云海学长为了探究神秘的物理现象，从高楼抛下一个弹球 但由于高空抛物被警察抓了，请各位小伙伴完成他未完成的心愿

小球从某一高度落下，每次落地后反跳回原来高度的一半，再落下。

输入弹球的初始高度 $M$ 和回弹次数 $N$ ，输出弹球第 $N$ 次回弹后的高度 和达到该高度后所经过的距离(保留两位小数)

( $0 < M < 2000$ ,  $0 < N < 50$ )

样例输入

1000 5

样例输出

31.25

2906.25

## 4. 嵌套循环

1. 使用程序打印九九乘法表

样例输出：

1 \* 1 = 1

1 \* 2 = 2      2 \* 2 = 4

1 \* 3 = 3      2 \* 3 = 6      3 \* 3 = 9

...

2. 求素数

设计一个程序，输入 $N$ ，输出 $0 \sim N$ 之间的所有素数

( $0 < N < \text{MAX\_INT}$ )

样例输入：

100

样例输出：

2

3

5

7

11

13  
17  
19  
23  
29  
31  
37  
41  
43  
47  
53  
59  
61  
67  
71  
73  
79  
83  
89  
97

### 3. 小游戏：猜数字

输入整数 $n$ ( $0 < n < 100$ )，想让程序猜到的值

如果程序没有猜中，则提示太大/太小 并让程序接着猜测

程序通过二分法不断缩小猜测范围，直到猜中 $n$

例如 $n = 39$

程序第一次猜测50，判断数值太大

第二次猜测25，判断数值太小

第三次猜测37 判断数值太小

43 大

40 大

38 小

39 正确

输入： $n$  表示被猜测值

输出：每行输出一个整数，表示程序当前猜测的数字

最终行输出猜测的次数

样例输入：

39

样例输出：

50

25

37

43

40

38

39

最终猜测了7次

### 4. 打印菱形

输入菱形的边长 $n$ ，打印一个对应大小的菱形

( $1 < n < 100$ )

样例输入：5

样例输出：

```
    *
   ***
  *****
 *****
*****
*****
*****
  ***
   *
```

## 5. 一维数组

### 1. 删除重复值

阿伟学长在开发一款游戏，他希望玩家已经获得的道具会降低爆率，避免玩家获得太多重复的道具，但玩家背包中道具是乱序且不确定的，为了减小判断的运算量，需要先对数据进行去重。

设计一个程序，输入数据数量 $N$ ，以及 $N$ 个整数 $N_i$ 。将该组数据去重后打印

( $1 < N < 100$ ,  $0 \leq N_i \leq \text{MAX\_INT}$ )

样例输入：

10

1 3 2 6 2 6 4 8 2 6

样例输出：

1 3 2 6 4 8

### 2. 数字环

有一个长度为 $n$ 的数字环，将每个数字往后移动 $m$ 位，使其成为一个新的数字环

输入：

第一行：整数 $n$  表示有 $n$ 个数字

第二行： $n$ 个整数 $N_i$ ，表示数字环的每个元素

第三行：整数 $m$  表示每个元素需要往后移动 $m$ 位

输出：移动后的新数字环

( $1 \leq m < n < 100$ ,  $\text{MIN\_INT} < N_i < \text{MAX\_INT}$ )

样例输入：

10

1 2 3 4 5 6 7 8 9 10

2

样例输出：

9 10 1 2 3 4 5 6 7 8

### 3. 合并有序数组

阿伟学长终于写好了游戏背包1和背包2的排序算法，但他发现将两个背包同时带上时，还得合并一下排序。阿伟学长懒得再写排序了，希望小伙伴能帮帮他完成背包的合并。

设计一个程序，输入两个有序数组，输出一个合并后的有序数组。

输入：

第一行：整数 $n, m$  表示数组1和数组2分别有 $n, m$ 个数字

第二行： $n$ 个整数 $N_i$ ，表示数组1的每个元素

第三行： $m$ 个整数 $M_i$ ，表示数组2的每个元素

( $n, m \leq 100, 0 \leq n_i, m_i \leq 1000$ )

输出：

合并后的新数组

样例输入：

5 6

1 3 5 7 9

1 3 5 6 7 8

输出：

1 1 3 3 5 5 6 7 7 8 9

#### 4. 排队枪毙

小约参与了犹太游戏，在这个游戏中最后活下来的人才能获得胜利。

游戏规则是这样的， $n$ 个玩家们围成一个圈，从第一个玩家开始报数

报到 $m$ 的人被枪毙，下一个人接着从1开始报数。

直到最后一个活下来的人获得胜利。

小约现在开始选编号了，已知参与的玩家数量为 $n$ ，被枪毙的数是 $m$ 。小约要选到几号位置才会活下来。

( $1 < m < n < 100$ )

设计一个程序，输入 $n$ 和 $m$ ，输出最后的胜利者

样例输入：10 3

样例输出：4

## 6. 多维数组

#### 1. 矩阵转置

输入一个数字构成的矩形，将矩形的值进行转置后打印

输入：

第一行 正整数 $n$ ( $1 < n < 10$ )，表示矩阵的边长

随后输入一个矩阵

输出：

转置后的矩阵

样例输入：

3

1 2 3

4 5 6

7 8 9

样例输出：

1 4 7

2 5 8

3 6 9

#### 2. 颈椎病治疗

最近云海学长一直对着电脑改bug，颈椎不舒服，希望各位小伙伴帮云海学长治治

提供一张图片，将图片旋转后再发给云海学长，这样学长看图的时候就需要歪着脖子，时间久了，颈椎病就治好了

输入一个数字构成的矩形，将矩形的值进行90度旋转后打印

输入：  
第一行 正整数 $n(1 < n < 10)$ ，表示矩阵的边长  
随后输入一个矩阵  
输出：  
90度旋转后的矩阵

样例输入：

```
3
1 2 3
4 5 6
7 8 9
```

样例输出：

```
7 4 1
8 5 2
9 6 3
```

### 3. 杨辉三角

输入 $n(1 < n < 10)$ ，打印高度为 $n$ 的杨辉三角

样例输入：5

样例输出：

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

### \*4. 包围圈

东东哥在一场军事演练中，需要对一片正方形森林进行排查，  
为了防止被包围，东东哥需要先排查外围才能进一步深入

输入：正整数 $n(1 < n < 10)$ ，表示森林的边长

输出：东东哥排查森林的顺序

样例输入：

4

样例输出：

```
1  2  3  4
12 13 14  5
11 16 15  6
10  9  8  7
```

## 7. 字符/字符串

### 1. 大小写转换

设计一个程序，输入一行字符串，将其中大写转为小写，小写转为大写。其余字符不变  
(字符串长度 $< 100$ )

样例输入：hELLO wORLD!

样例输出：Hello world!

### 2. 查找与替换

文本编辑器通常都拥有查找/替换的功能，该功能的快捷键是`ctrl+f`。

现在 需要你设计一个替换功能，输入原文 $S$  以及 $A, B$ 两个子串，用 $B$ 替换掉 $S$ 中的 $A$

输入：

第一行：字符串S  
第二行：子串A  
第三行：子串B  
(S长度<100, A,B长度<10)

样例输入：  
把你的心 我的心 串一串  
心  
鸡翅

样例输出：  
把你的鸡翅 我的鸡翅 串一串

### 3. 开挂很危险

最近网络盗号严重，张三又喜欢开外挂(大家不要学)  
于是张三打算出一个安全的密码，避免开外挂后被盗号  
请各位小伙伴帮忙，判定一下张三的密码是否安全

安全密码的判定：

1. 长度在8~16之间
2. 开头必须是大写
- 3\* 密码最少包含一个小写字母，一个数字，一个特殊符号  
特殊符号包括~!@#\$%\*

设计一个程序 输入密码字符串  
如果密码安全，打印true，否则打印false  
(字符串长度<100)

样例输入：  
Aa123!!!!  
样例输出：  
true

### 4. 密码加密

自从上次张三密码变复杂后，再也没有被盗过号  
但他觉得密码还是不够复杂，于是又进行了二次加密  
然后..他忘记二次加密后的密码了  
小伙伴赶紧帮他找找

张三的加密方式是这样的：  
对于每个字母和数字，按字母表/数字顺序往后挪5位  
(如果超过了 则从头开始，比如'z'转换后变成'e'，'9'转换后变成'4')  
符号不变

如密码：AaZz09!!!  
加密后：FfEe54!!!

设计一个程序，输入加密前的密码 打印加密后的密码  
(密码长度 <= 16)

样例输入：AaZz09!!!  
样例输出：FfEe54!!!



## 8. 函数

1. 根据以下要求，实现一套登录功能

已知有全局变量

```
int ids[] = {10001,10002,10003,10004};
String names[] = {"张三", "李四", "王五", "赵六"};
String passwords[] = {"aaaaa","bbbbbb","ccccc","dddddd"};
int uNum = 4;
```

其中 每个下标对应一套用户数据(id, 名称, 密码)

如下标[1]: 对应10002, 李四, bbbbbb

根据以下的函数声明和注释，写出相应的函数实现

```
/*
    功能：根据id 查询用户是否存在，如果存在返回用户名， 如果不存在返回空
    参数：
        uid: 用户id
    返回值：
        如果用户存在，返回用户名。
        如果用户不存在，返回NULL
*/
String selectUserById(int uid);

/*
    功能：根据id 查询用户密码，如果存在返回用户密码， 如果不存在返回空
    参数：
        uid: 用户id
    返回值：
        如果密码存在，返回密码。
        如果密码不存在，返回NULL
*/
String selectPassById(int uid);
-----
/*
    功能：传入用户id和密码，根据上面两个函数(selectUserById, selectPassById)来获取相应用户数据，并判断是否登录成功
        传入用户id 查询用户名是否存在，并获取用户密码
        如果用户存在 则判断密码是否正确
    参数：
        uid: 用户账户
        password: 用户密码
    返回值：
        如果账号不存在，返回1
        如果密码错误，返回2
        如果登录成功，返回0
*/
int login(int uid, String password);
-----
/*
    功能：提示用户输入账号密码，根据login函数判断是否登录成功，
    如果登录成功提示正在进入首页
    如果登录失败
        密码错误：提示密码错误，并让用户重新登录
```

账号不存在：提示账号不存在，并提示正在进入注册界面

参数：无

返回值：无

\*/

void showLoginPage();

样例输入：

10001

aaaaa

样例输出：

登录成功，正在进入首页

样例输入：

10001

aaaab

样例输出：

密码错误，请重新登录

提示：先输入数字再输入回车，可能会出现回车符号被next或者nextLine接收的情况，导致密码无法被接收，

可以多写一个nextLine接收回车来解决。

对应知识点：键盘缓冲区

2. 上面有用虚线将函数划分为3种，这三种函数有哪些区别(可以百度查询"三层架构")。根据这种思路实现一套注册功能。

## 9. 面向对象

实现该学生管理系统

class Person:

抽象类，具有人的基本属性和方法

属性：

成员变量：年龄(age)，姓名(name)，性别(sex)

静态变量：count：用于统计有多少Person对象

方法：

构造方法：传入年龄，性别和姓名。将对应数据进行修改

抽象方法show；打印该对象的信息

class Student:

继承自Person类

属性：

成员变量：学号(sid)

静态变量：count：用于统计有多少Student对象

方法：

构造方法：传入年龄，性别和姓名，将对应数据进行修改。并通过hashCode()生成学号

equals方法：传入Person类，如果是Student类且学号相等，判定为同一人

hashCode方法：根据年龄，性别和姓名，生成尽量独一无二的学号

show方法：打印个人信息

class Worker:

继承自**Person**类

属性:

成员变量: 工号(wid)

静态变量: **count**: 用于统计有多少**worker**对象

方法:

构造方法: 传入年龄, 性别和姓名, 将对应数据进行修改. 并通过**hashCode()**生成工号

**equals**方法: 传入**Person**类, 如果是**worker**类且学号相等, 判定为同一人

**hashCode**方法: 根据年龄, 性别和姓名, 生成尽量独一无二的工号

**show**方法: 打印个人信息

**class Management:**

存储**Person**及其子类

属性:

自定义

方法:

**add**方法: 传入**Person**类, 如果该数据重复 则返回**false**, 否则将数据存入系统并返回**true**

**delStudent**: 传入学生**id**, 删除对应数据. 如果删除失败返回**false** 否则返回**true**

**delWorker**: 传入工人**id**, 删除对应数据. 如果删除失败返回**false** 否则返回**true**

**showStudent**: 打印系统中所有的学生数据

**showWorker**: 打印系统中所有的工人数据

**showStat**: 打印系统中学生人数, 工人人数

调用样例:

```
static final boolean MAN = true;
static final boolean WOMAN = false;
public static void main(String[] args) {
    Management mana = new Management();

    // add测试
    System.out.println(
        (mana.add(new Student(18, "张三", MAN)) &&
        mana.add(new Student(19, "李四", WOMAN)) &&
        mana.add(new Worker(20, "王五", MAN)) &&
        mana.add(new Worker(21, "赵六", WOMAN)) &&
        !mana.add(new Worker(21, "赵六", WOMAN))) ? "测试成功" : "测试失败"
    );

    //del及hashCode测试
    int tempId = new Worker(21, "赵六", WOMAN).hashCode();
    System.out.println(
        (mana.delStudent(tempId) ||
        !mana.delWorker(tempId)) ? "测试失败" : "测试成功"
    );

    // count测试
    System.gc();
    try { Thread.sleep(1000); } catch (InterruptedException e)
    {e.printStackTrace();}
    System.out.println(
        (Student.count == 2 &&
        worker.count == 1 &&
        Person.count == 3) ? "测试成功": "测试失败"
    );

    // 打印
```

```
mana.showStudent();
mana.showWorker();
mana.showStat();
}
```

样例输出：

测试成功

测试成功

测试成功

系统学生数据：

学号：213717737 姓名：张三 年龄：18 性别：男

学号：213650565 姓名：李四 年龄：19 性别：女

系统工人数据：

工号：1697280996 姓名：王五 年龄：20 性别：男

系统统计数据：

学生人数：2

工人人数：1

## 10. 集合

1. 实现下列的add方法，使得链表head是一个有序链表

```
class MyNode{
    MyNode next;
    int data;
    void add(MyNode head, int val){
        // 在此处实现代码
    }
}
```

样例调用

```
public static void main(String[] args) {
    MyNode head = new MyNode();
    head.add(head, 5);
    head.add(head, 1);
    head.add(head, 3);
    head.add(head, 2);
    head.add(head, 4);
    MyNode temp = head.next;
    while(temp != null){
        System.out.print(temp.data);
        temp = temp.next;
    }
}
```

样例输出

12345

2. 通过ArrayList的结构思路，实现该接口：

```

interface MyList<E>{

    /**
     * 获取元素数量
     * @return: 集合中存储的元素数量
     */
    int size();

    /**
     * 判断集合中是否有存储元素
     * @return: 如果有存储元素 返回true 否则返回false
     */
    boolean isEmpty();

    /**
     * 设置指定下标的元素值
     * @param index: 指定下标
     * @param e: 修改后的元素
     * @return: 修改前的元素, 如果下标越界 返回null
     */
    E set(int index, E e);

    /**
     * 移除指定元素
     * @param index: 被移除的元素下标
     * @return: 被移除的元素, 如果下标越界 返回null
     */
    E remove(int index);

    /**
     * 添加一个元素
     * @param e: 被添加的元素
     * @return: 如果添加成功, 返回true 否则返回false
     */
    boolean add(E e);

    /**
     * 获取指定下标的元素
     * @param index: 指定元素的下标
     * @return: 指定的元素, 如果index越界 返回null
     */
    E get(int index);
}

```

样例调用:

```

MyList<String> list = new MyArrayList();
for(int i = 0; i < 11; i++){
    list.add(i+"");
}
System.out.println(list.remove(5));
System.out.println(list.set(5,"6"));
System.out.println(list.add("11"));
for(int i = 0; i < 15; i++){
    System.out.print(list.get(i) + "\t");
}

```

样例输出：

```
5
6
true
0 1 2 3 4 6 7 8 9 10 11 null null null null
```

## \*1. swing

1. 设计一个绘图程序。

1. 该程序是一个高宽为800\*600的白色界面。
2. 通过按下a/d可以切换颜色，有红，粉红，黑，白四种颜色。
3. 鼠标按下时，移动鼠标(笔刷)可以在白色背景上留下对应颜色的印记。
4. 按下q/e可以调整笔刷大小，笔刷是最小为1,最大为25(半径)的圆形。

输入：画一只佩奇，发到图灵院群 问大家画的好不好看

输出：[图片1]

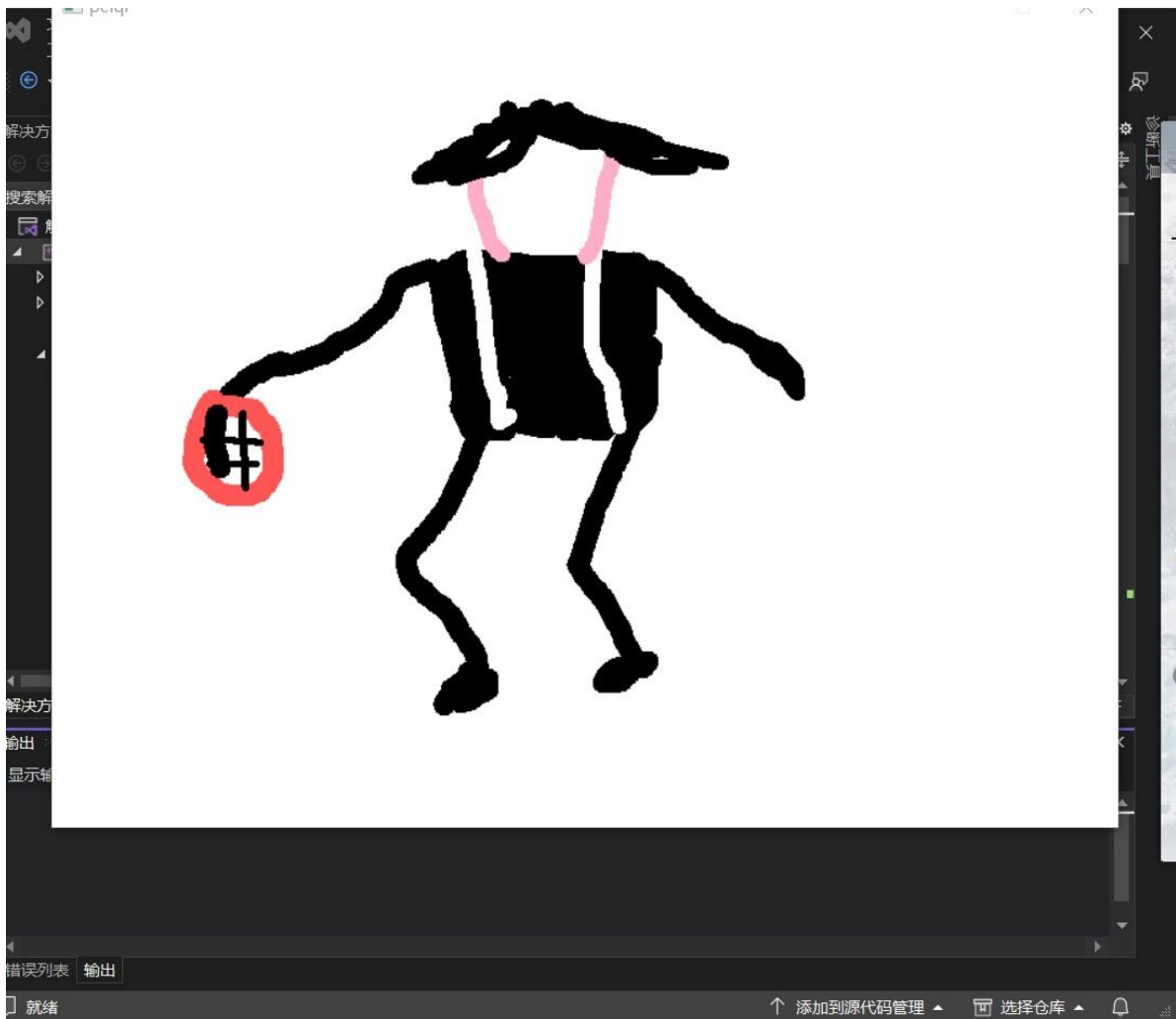
2. 设计一个计算器程序，该程序同时可以用键盘进行操作(数字，符号，删除)。

注意：要求界面使用布局管理器+组件 而非贴图

输出：[图片2]







## \*2. 文件流

1. 制作一个排行榜，每次修改其数据时，都会在文件中进行对应操作。

该排行榜可以展示名次，昵称，积分三个属性

每次用户可以输入昵称和积分，来向排行榜中加入数据。

(昵称长度<10，积分<10000)

加入数据时，如果排行榜中存储数量超过10个，则删除最后一名数据。



样例输入：

张三 100

样例输出：

名次	昵称	积分
1	测试1	1001
2	测试2	100
3	张三	100
4	测试3	10