

北京科技大学 第七届acm-icpc校内选拔赛 暨第一届CCCC-GPLT校内选拔赛 宣讲会

LETTERS



IBM | event sponsor



 [Start With A Problem](#)

[About ACM-ICPC](#) 

 [About CCCC-GPLT](#)

[About LETTers](#) 

 [Something To Share](#)

[Our Contest](#) 

 [How To Start](#)

[Try Some More Problems](#) 

Start With A Problem

CrazyX

Array without 11

- **Description**

Now give you the **length m ($1 \leq m \leq 90$)** of an array which **consists of 1&0**. (It means elements of the array will only be 1 or 0.)

But we don't like "1" with "1". So you need to count how many different arrays of this length **without "11"**.

Example:

$m=2$

suitable arrays: **10 01 00**, so the answer is **"3"**

("11" is illegal, do not count it. "10" and "01" are considered different.)

Array without 11

- **Input**

The first line is a positive integer: n for number of cases.

For each case, the first line is a positive integer: m

- **Output**

For each case, output the answer in a single line.

- **Sample Input**

3

1

2

5

- **Sample Output**

2

3

13

如何解题

- 题目大意
 - 长度为m的01串，不能包含“11”，问有多少个符合条件。
- 小范围内的答案
 - 样例中已经给出了 $\text{ans}[1]=2, \text{ans}[2]=3, \text{ans}[5]=13$,
 - 手推一下 $n=3$ 和 $n=4$ 的答案， $\text{ans}[3]=5, \text{ans}[4]=8...$
 - **2 3 5 8 13**... 似乎有 $\text{ans}[i]=\text{ans}[i-1]+\text{ans}[i-2] (i \geq 3)$
- 好像有点意思了，我们可以写个暴力搜索的程序，来得出稍微大一点范围内的规律

暴力程序

```
int ans=0,n;
void dfs(int i,int pre){
    if(i==n){
        ans++;
        return;
    }
    if(pre==1){
        dfs(i+1,0);
    }else if(pre==0){
        dfs(i+1,1);
        dfs(i+1,0);
    }
}
```

- 当n=40时，已经明显不能在规定的1s时间内得出答案了

- 但是

- 答案数列为:

2, 3, 5, 8, 13, 21, 34, 55,
89, 144 ...

$ans[i]=ans[i-1]+ans[i-2]$

规律

- $\text{ans}[i] = \text{ans}[i-1] + \text{ans}[i-2]$
- 为什么？
- 假设现在已经知道了 $\text{ans}[i]$ 和 $\text{ans}[i+1]$ ，要计算 $\text{ans}[i+2]$ 。
- 可以发现，在 $i+1$ 的所有情况中，最后添加一个 0，在 i 的所有情况中，最后添加一个 01，恰好就是 $\text{ans}[i+2]$

更清晰的动态规划的思想

- $dp[i][0]$ 表示长度为 i ，结尾为0的方案数
- $dp[i][1]$ 表示长度为 i ，结尾为1的方案数
 - $ans[i] = dp[i][0] + dp[i][1]$
- 并且有
 - $dp[i][0] = dp[i-1][0] + dp[i-1][1]$
 - $dp[i][1] = dp[i-1][0]$

一些其他的注意事项

- $m=90$ 时答案为7540113804746346429 , $7e18$
- 注意`int`的取值范围在 $2^{31}-1$ 以内 , 大概 $2e9$ 左右 , 当超出范围时请使用`long long` ($2^{63}-1$)数据类型 (Java中用`long`)

题目的一种变式

- 数据范围增大 $m \leq 10^6$
- 输出答案对 10^9+7 取模

$m \leq 10^6$, 答案对 10^9+7 取模

- 首先需要知道一个东西
- 取模运算满足
 - $(a+b)\%M=(a\%M+b\%M)\%M$
 - $(a-b)\%M=(a\%M-b\%M+M)\%M$
 - $(a*b)\%M=((a\%M)*(b\%M))\%M$
- 所以对上述要求，依旧采取之前的动态规划思路或者递推思路，中间运算过程加上取模即可

考虑更大的数据范围？

- $m \leq 10^9$ ，输出答案对 10^9+7 取模
- 需要用到一些线性代数的知识...
- 已知 $a[1], a[2], a[i] = a[i-1] + a[i-2] (i \geq 3)$ ，求 $a[m]$;
- 构造矩阵 $A = \begin{bmatrix} a[i-1] & a[i-2] \end{bmatrix}$, $B = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$
 - $B \cdot A = [a[i-1] + a[i-2], a[i-1]] = [a[i], a[i-1]]$
 - $A[0] = [a[2], a[1]]$ 那么 $B^m \cdot A[0] = [a[m+1], a[m]]$
- 其中 B^m 可以在 $\log(m)$ 的时间内求出
- 具体需要用到快速幂的思想对矩阵进行快速幂

About ACM-ICPC

Eopxt

ACM-ICPC简介



- ACM国际大学生程序设计竞赛（英文全称：ACM International Collegiate Programming Contest，简称ACM-ICPC或ICPC）是由美国计算机协会（ACM）主办的，一项旨在展示大学生创新能力、团队精神和在压力下编写程序、分析和解决问题能力的年度竞赛。经过四十多年的发展，ACM国际大学生程序设计竞赛已经发展成为全球最具影响力的大学生程序设计竞赛。
- 赛事目前由IBM公司赞助。

国内发展历史

- 中国大陆高校自1996年开始参加此项赛事的亚洲区预赛
- 1996~2001年，上海大学连续举办6届区域赛（Regional）
- 之后在大陆地区设置了多个赛点，由各大学轮流主办地区性竞赛
- 到2016年止，大陆地区已经举办了二十一届了
- 大陆地区还承办过两次全球总决赛（World Final）：
 - 2005年4月，上海交通大学，第29届ACM国际大学生程序设计竞赛全球总决赛
 - 2010年2月，哈尔滨工程大学，第34届ACM国际大学生程序设计竞赛全球总决赛

<http://acm.hdu.edu.cn/region/index.php?year=2016>

区域赛与全球总决赛

- 区域赛网络预选赛 (Regional Online)
 - 每年9月份
 - 选拔参加现场赛的队伍
 - 中国特色
- 区域赛现场赛 (Regional Onsite)
 - 每年9~12月份
- 全球总决赛 (World Final)
 - 每年2~5月份

简要规则

- 3名选手共用1台电脑的团体比赛
- 5小时内挑战10道左右的问题
 - 组合数学、数论、计算几何、博弈论、动态规划等
- 使用C/C++/JAVA编写程序，在时限内通过所有测试数据
- 通过题数更多，用时更少的队伍排在前面
- 每通过一道题会获得一只代表该题的气球
- 团队配合、编程功底、心理素质

ACM给我们带来什么？

- 强大的自学能力
- 严密的逻辑思维
- 程序实现能力
- 淡定从容的心态
- 开阔的视野
- 真挚的友谊
- 优秀的学习榜样

About CCCC-GPLT

Tom

CCCC 中国高校计算机大赛

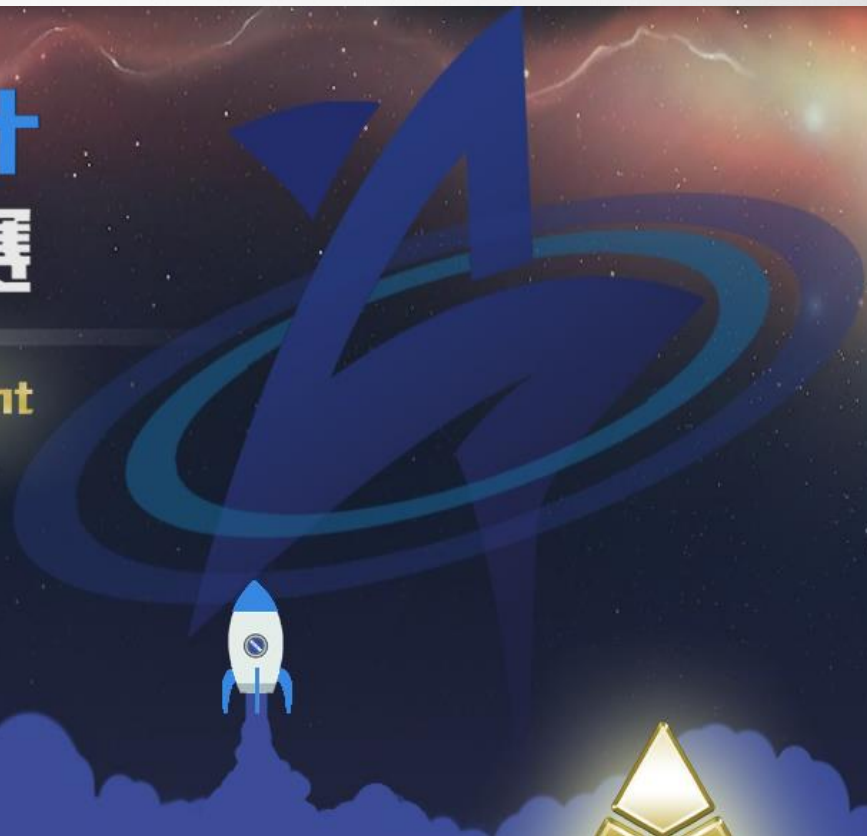
- 团体程序设计天梯赛
 - Group Programming Ladder Tournament
- 大数据挑战赛
 - Big Data Challenge
- 移动应用创新赛
 - Mobile Application Innovation Contest

GPLT 团体程序设计天梯赛



团体程序设计
天梯赛

Group Programming Ladder Tournament



CCCC-GPLT VS ACM

- 中文题目
- 初赛3小时，决赛2.5小时
- 15道题
- 按照测试点给分

1. 竞赛题目分 3 个梯级：

- 基础级设 8 道题，其中 5 分、10 分、15 分、20 分的题各 2 道，满分为 100 分；
- 进阶级设 4 道题，每道题 25 分，满分为 100 分；
- 登顶级设 3 道题，每道题 30 分，满分为 90 分。

2. 参赛队员必须独立按照严格的输入输出要求提交每一题的解题程序。程序须经过若干测试用例的测试，每个测试用例分配一定分数。每题的得分为通过的测试用例得分之和；整场比赛得分为各题得分之和。可以反复提交，取最高分，提交错误不扣分。

3. 参赛队员可以在比赛中的任何时刻尝试解决任何梯级的题目。但只有当一支队伍的基础题总分超过 800 分时，其本队进阶部分的题目分数才被判为有效。只有当其进阶题总分超过 400 分时，其本队登顶部分的题目分数才被判为有效。

4. 在进阶和登顶阶段最先完整获得其中任一题分数的前 5 位队员，分别依次获得 50、40、30、20、10 分“先锋奖励”。该奖励分数计入个人总分。当团队进阶或登顶成功后，也计入该团队的有效分。

About LETTers

John

Something To Share

nbybyyby

Our Contest

Martian

时间表

- 报名截止日期：2016年11月21日（含）
- 报名方式
 - 登录本科教学网旧版教务管理系统 竞赛报名
- 预赛：
 - 2016年11月27日（**第11周 周日**）**13:00~16:00**
- 决赛：
 - 2016年12月03日（**第12周 周六**）**13:00~17:00**
- 上述时间如有修改会及时通知

预赛

- 预赛采取**网络赛**的形式。
- 参赛选手可在校内任何地点**连接校园网**访问相应网站竞赛系统参加比赛。（可以在宿舍打，图书馆也可以。其它任何地方只要能登录校园网就可以。）
- 预赛的网址会在**公邮**（ustbacm2016@ourmail.cn）里面公布。
- 这段时间多关注公邮。里面会发比赛地点以及相关文件。

预赛注意事项

- 如果没有意外，预赛比赛时长为**3小时**。
- 在比赛过程中，关注比赛网站中的**通知区域**，我们会在那里会发一些通知。
- 在比赛网站中会有**讨论区**。如果对试题描述中的不明确或错误的部分有问题，可以在比赛的讨论区向工作人员提问。

这里是通知区域，比赛时如果有通知将会在此滚动显示

通知区域

Problem ID: User: Language: All Result: All

RunID	User	Problem	Result	Memory	Time	Language	Code Length	Submit Time
1203	admin	1042	Accepted	1696	0	C++	693 B	2016-11-17 22:47:51
1202	3435973836	1042	Accepted	1696	0	C++/Edit	583 B	2016-11-17 22:46:31
1201	admin	1042	Compile Error	0	0	C++	694 B	2016-11-17 22:44:28
1200	3435973836	1042	Accepted	1700	0	C++/Edit	1762 B	2016-11-17 22:43:21

讨论区

Contest1002 - Sample Contest

Start Time: 2016-11-16 10:45:00 End Time: 2016-11-16 15:45:00

Current Time: 2016-11-18 15:43:48 Status: Ended Public

[\[Status\]](#) [\[Standing\]](#) [\[Statistics\]](#)

Problem ID	Title	Source
1008 Problem A	Go home	
1006 Problem B	Collect coins	
1007 Problem C	Learning math	
1009 Problem D	Sum	
1010 Problem E	Lasercat numbers	

预赛规则

- 允许进行搜索（禁止拷贝代码，会有代码查重）
- 如发现有涉及抄袭的代码，将**取消所有相关选手的比赛成绩**。
 - 比赛期间请保管好自己的用户名和密码。
 - 登录之后**及时修改默认密码**。
 - 不要向别人透露自己的用户名和密码。

关于晋级

- 通过预赛的选手可参加现场决赛。
- 预赛结果将在比赛结束后三天内以统一的方式告知所有参赛选手。

关于排名

[Home](#)[Web Board](#)[ProblemSet](#)[Standing](#)[Status](#)[Statistics](#)

这里是通知区域，比赛时如果有通知将会在此滚动显示

Contest RankList -- Sample Contest

[Download](#)

Rank	User	Nick	Solved	Penalty	A	B	C	D	E	F
Winner	41458011	李志远	2	04:46:07			02:39:38(-5)	(-1)	00:26:29	
2	Smile	Smile	2	05:13:24	00:16:20(-1)			03:57:04(-2)		(-8)
3	crazyX	crazyX	1	01:29:14	01:29:14					
*	Star	*Star	1	07:08:12		(-1)		04:08:12(-9)		

决赛

- 决赛采取**现场赛**的形式。
- 比赛地点会在逸夫楼或机电楼的机房
 - （如逸夫楼804、904、机电楼303）
 - 目前地点待定，具体地点会在公邮（ustbacm2016@ourmail.cn）里通知。

决赛注意事项&规则

- 决赛时长为**4小时**。如有变更会现场告诉大家。
- 决赛须**携带本人学生证**准时到场。
- 决赛过程中的通知会在比赛网站上通知大家，也会在现场给大家通知。
- 决赛时可以举手示意工作人员问问题。
- 决赛时**可以带 unlimited 纸质资料，不可以携带任何电子资料、电子产品**
- 决赛结束后会现场颁奖，获奖的同学不要提前离场

评测环境

- 操作系统
 - Ubuntu 14.04
- 编译器
 - C/C++ : GNU GCC/G++ 4.8.4
 - Java : OpenJDK 1.7
- 编译参数
 - 详见OJ的FAQ页

决赛选手环境

- 操作系统

- Windows 7
- Ubuntu 14.04

- 语言

- C
- C++
- Java

Windows 7

C/C++

- IDE:
 - Eclipse CDT
 - Code::Blocks
 - Dev Cpp
- 编译器:
 - MinGW

Java

- IDE :
 - Eclipse
- 编译器 :
 - JDK 1.7

Ubuntu

C/C++

- IDE:
 - Eclipse CDT
 - Code::Blocks
- 编译器:
 - GNU GCC/G++

Java

- IDE :
 - Eclipse
- 编译器 :
 - OpenJDK 1.7

How To Start

YuanZhaolin

入门推荐

• 书

- 挑战程序设计 (大白)
- 算法竞赛入门经典
- 红书系列中的算法与实现

• 在线评测系统

- HDU (acm.hdu.edu.cn)
- POJ (poj.org)
- Uva (uva.onlinejudge.org)
- CodeForces (codeforces.com)
-



多看
多想
多写

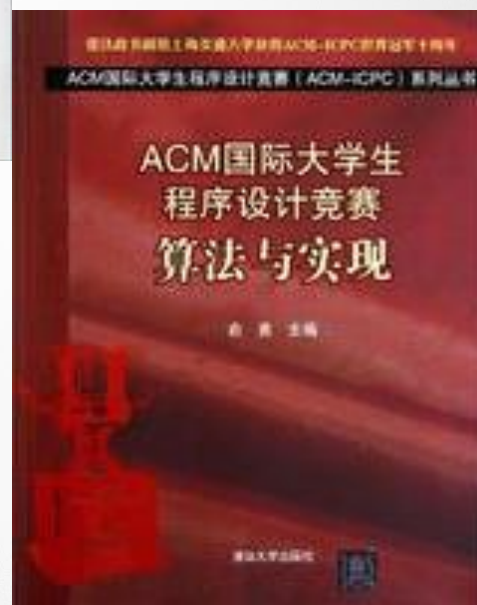


人民邮电出版社
POSTS & TELECOM PRESS

挑战 程序设计竞赛

(第2版)
世界顶级程序设计高手的经验总结
ACM-ICPC全球总冠军 巫泽俊 主译

【中】 杨叶松 陈伟 周天一 王川 等 译
原译 陈伟 李博 译
陈 威 王 陆 等



数据输入输出

- 谨记的要点：
 - 不要乱加其他东西；
 - 不要忘掉要求的東西；
 - 不要全读完一起算；
 - scanf/printf大法好；

输入Input

1. 单组输入（简单但少见）

- 只有一组测试数据，输出完结果程序结束（return 0）
- `#include "stdio.h"`
`int main()`
`{`
 `int a, b;`
 `scanf("%d%d", &a, &b);`
 `printf("%d\n", a + b);`
 `return 0;`
`}`

Input

Two integer a,b ($0 \leq a, b \leq 10$)

Sample Input

1 2

输入Input

2. 多组输入但没有明确指出输入什么时候结束

- 有多组测试数据，但不告诉你到底有多少组！
- 技巧：只需要背下来三个字母，他们叫做 EOF.

Input

```
int main()
{
    int a, b;
    while (scanf("%d%d", &a, &b) != EOF)
    {
        printf("%d\n", a + b);
    }
    return 0;
}
```

The input will consist of a series of pairs of integers a and b, separated by a space, one pair of integers per line.

Sample Input

```
1 5
10 20
```


输入Input

3. 多组输入，并且告诉你有多少组

- 第一行准确告诉你有多少组
- 技巧：加个变量即可

```
int main()
{
```

```
    int T, a, b;
```

```
    scanf("%d", &T);
```

```
    for(int ca=1;ca<=T;ca++)
```

```
    {
```

```
        scanf("%d%d", &a, &b);
```

```
        printf("%d\n", a + b);
```

```
    }
```

```
    return 0;
```

```
}
```

Input

Input contains an integer N in the first line, and then N lines follow.

Each line consists of a pair of integers a and b, separated by a space, one pair of integers per line.

Sample Input

```
2
1 5
10 20
```

输入Input

4. 多组输入，以特定值为结尾

- 没有具体组数，而以某些变量输入的值作为标志代表结束。
- 技巧：加个if 和 break就好啦！

```
int main()
{
    int a, b;
    while (scanf("%d%d", &a, &b) != EOF)
    {
        if(a==0&&b==0) break;
        printf("%d\n", a + b);
    }
    return 0;
}
```

Input

Input contains multiple test cases.
Each test case contains a pair of integers a and b, one pair of integers per line.
A test case containing 0 0 terminates the input and this test case is not to be processed.

Sample Input

```
1 5
10 20
0 0
```

输出Output

1. 每组case一个数

- 每组测试数据答案仅为一个数、每组答案占一行。

```
int main()
{
    int a, b;
    while (scanf("%d%d", &a, &b)!=EOF)
    {
        if(a==0&&b==0) break;
        printf("%d\n", a + b);
    }
    return 0;
}
```

Output

Output a+b

Sample Input

1 2

Sample Output

3

输出Output

2. 每组case多个数

- 每组测试数据答案为多个数，用空格或回车隔开。
 - 两种情况

```
for(int i=0;i<n;i++)  
    printf("%d ",ans[i]); // 每个数后都有空格  
printf("\n");
```

```
for(int i=0;i<n;i++)  
{  
    printf("%d",ans[i]);  
    if(i!=n-1) printf(" "); // 最后一个数后没有空格  
}  
printf("\n");
```

校赛中都可以
其余OJ上看情况

输出Output

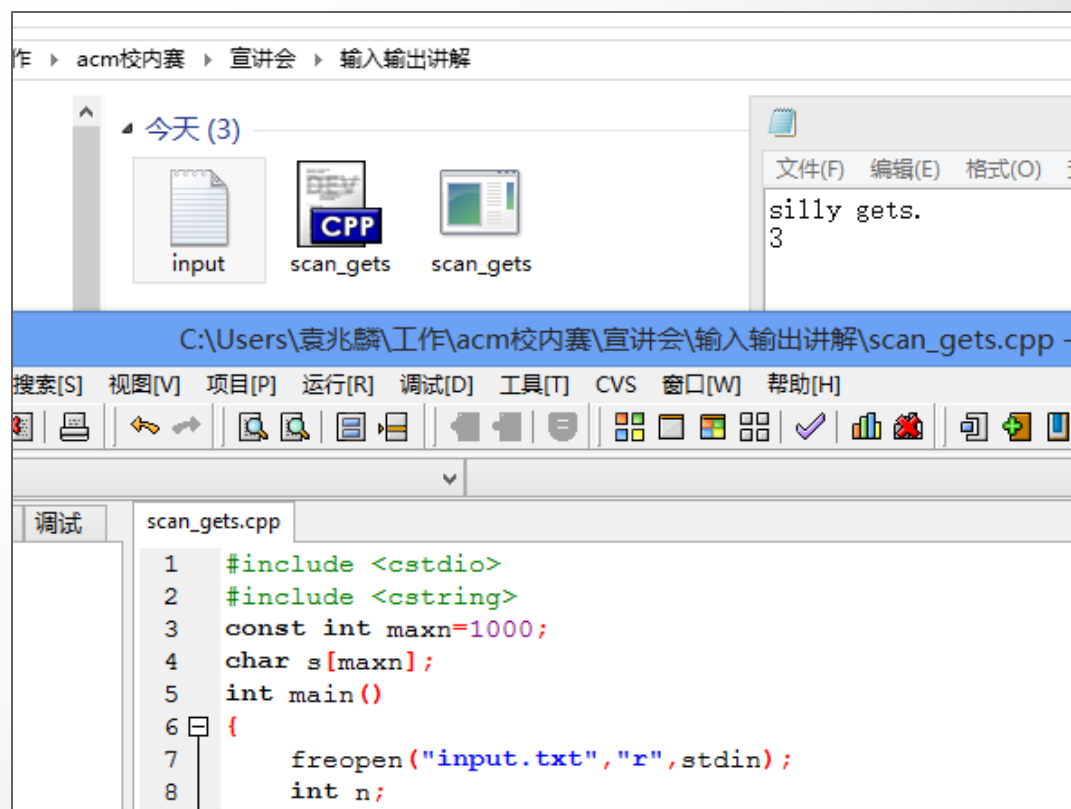
3. 每组case之间留空行

- 要求每组之间数据要留一空行。
- 技巧：除第一组外每组输出一个空行

```
int main()
{
    int a, b, T=0;
    while (scanf("%d%d", &a, &b) != EOF)
    {
        if(T++!=0) printf("\n");
        printf("%d\n", a + b);
    }
    return 0;
}
```

文件重定向

- `freopen("data.in","r",stdin);`
- `freopen("data.out","w",stdout);`
- 方便本地测试而已
- 提交代码不要加



提交后OJ返回的结果

- **Pending & Compiling & Running** : 评判程序，请耐心等待。
- **Accepted** : 恭喜！本题通过！
- **Compile Error** : 简称“CE”，代码编译错误。
- **Wrong Answer** : 简称“WA”，提交的代码运行后输出的答案错误。
- **Presentation Error** : 简称“PE”，输出格式错误。
- **Time Limit Exceed** : 简称“TLE”，运行时间超出题目限制。
- **Runtime Error** : 简称“RE”，运行时发生崩溃或者错误。
- **Memory Limit Exceed** : 简称“MLE”，程序占用内存空间超出题目限制。
- **Output Limit Exceed** : 简称“OLE”，程序输出内容过多。

其他注意事项

1. `scanf("%d%d",&a,&b);`在%d之间不加空格、逗号之类的东西！

不要写`scanf("%d %d",&a,&b);` `scanf("%d,%d",&a,&b);`

2. `scanf`与`gets`的区别

	<code>gets</code>	<code>scanf</code>
开始位置	直接开始	下一个非空格、非换行处
结束为止	下一行行首	下一个空格符、换行符
过程内容	一直到行尾换行符处(可能包含空格)	到最近空格符或者换行符(不会包括空格)

scanf与gets的区别

```
#include <stdio.h>
#include <string.h>
const int maxn=1000;
char s[maxn];
int main()
{
    freopen("input.txt","r",stdin);
    int n;
    scanf("%d",&n);
    gets(s);
    printf("%d\n",n);
    printf("%s\n",s);
    return 0;
}
```

C:\Users\袁兆麟\学习\竞赛

3

Process exited with return value 0
Press any key to continue . . .

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

3
silly gets.

scanf与gets的区别

```
#include <stdio.h>
#include <string.h>
const int maxn=1000;
char s[maxn];
int main()
{
    freopen("input.txt","r",stdin);
    int n;
    scanf("%d",&n);
    getchar();
    gets(s);
    printf("%d\n",n);
    printf("%s\n",s);
    return 0;
}
```

C:\Users\袁兆麟\学习\竞赛\...

3
silly gets.

Process exited with return value 0
Press any key to continue . . .

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

3
silly gets.|

scanf与gets的区别

scan_gets.cpp

```
1  #include <stdio.h>
2  #include <string.h>
3  const int maxn=1000;
4  char s[maxn];
5  int main()
6  {
7      freopen("input.txt", "r", stdin);
8      int n;
9      gets(s);
10     scanf("%d", &n);
11     printf("%d\n", n);
12     printf("%s\n", s);
13     return 0;
14 }
```

C:\Users\袁兆麟\学习\竞

3
silly gets.

Process exited with return value 0
Press any key to continue . . .

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

silly gets.|
3

scanf与gets的区别

```
scanf_get.cpp
1  #include <stdio.h>
2  #include <string.h>
3  const int maxn=1000;
4  char s[maxn];
5  int main()
6  {
7      freopen("input.txt", "r", stdin);
8      int n;
9      gets(s);
10     getchar();
11     scanf("%d", &n);
12     printf("%d\n", n);
13     printf("%s\n", s);
14     return 0;
15 }
```

C:\Users\袁兆麟\学习\竞赛\

66
silly gets.

Process exited with return value 0
Press any key to continue . . .



文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

silly gets.|
3

其他注意事项

- 3. Java的主类名必须是**Main**
- 4. C/C++的主函数最好是**int main()**,返回值为**return 0;**
- 5. 注意题目中整数数据规模，超过int类型上下限的，C/C++中可使用**long long**类型，Java中可使用**long**类型。

常见算法

- 贪心；
- 动态规划；
- 数学相关：组合数、数论、概率论、离散数学
- 图论；
- 数据结构；
-

Try Some More Problems

Martian

Come On!

We're waiting for you!