

读入技巧&文件操作

几种读入方式

1. 1.最简单的输入:

例如: 给定一个n,再输入n个数。

C:

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

```
for(int i = 0; i < n; i++) scanf("%d",&a[i]);
```

C++:

```
cin >> n;
```

```
for(int i = 0; i < n; i++) cin >> a[i];
```

几种读入方式

2.一次运行，有T组数据。

C:

```
int T;  
scanf("%d",&T);  
while(T--)  
{coding...}
```

C++:

```
int T;  
cin >> T;  
while(T--)  
{coding...}
```

几种读入方式

3.一次运行，有多组数据，直到读至输入文件末尾（EOF）为止

- 例如：输入a,b,输出a+b,多组数据
- C:
- `int a,b;`
- `while(scanf("%d",&a,&b) != EOF) printf("%d\n",a+b);`
- `int a,b;`
- `while(~scanf("%d",&a,&b)) printf("%d\n",a+b);`

几种读入方式

C++:

```
int a,b;
```

```
while(cin >> a >> b)
```

```
cout << a + b << endl;
```

- 注1: scanf函数返回值就是读出的变量个数, 如果一个都没有, 则返回值是-1。EOF是一个预定义的常量, 等于-1。
- 注2: 在调试程序时, 键盘输入的数据, 用CTRL-Z (即按住CTRL键不放, 再按下Z) 组合作为输入结束, 此谓键盘输入设备的“文件末尾”。

几种读入方式

4.输入不说明有多少组数据，但以某个特殊输入为结束标志。

例如：输入a,b,输出a+b,输入有多组数据，以0 0 结束。

C:

```
int a,b;
while(scanf("%d") && (a || b))
printf("%d\n",a+b);
```

C++:

```
int a,b;
while(cin >> a >> b && (a || b)) printf("%d\n",a+b);
```

字符读入

- 1.读入单个字符
- C++:
- `cin >> c;` //可以跳过空格和换行
- C:
- `scanf("%c",&c);`

因为%c会读入空格和换行

所以读入字符时要清除缓存区内容,或者吃掉空格和换行。

在读入字符前加入`fflush(stdin)`或者`getchar()`;

字符读入

- 2.读入字符串:

- C++:

- C++中有一个string类型，可以直接读入

- 例如:

```
string str;
```

```
cin >> str;
```

- C:

- char str[100]

```
scanf("%s",str); //遇到空格或者换行结束  
gets(str); //读入一行，即遇到换行结束
```


gets() 和scanf() 函数的区别

- gets可以接收空格；而scanf遇到空格、回车和Tab键都会认为输入结束，所有它不能接收空格。

```
char str[15]; gets(str); /*遇到回车认为输入结束*/
```

```
scanf("%s",str); /*遇到空格认为输入结束*/
```

所以在输入的字符串中包含空格时，应该使用gets输入。

读入优化

1.简单优化

a)只用scanf()、printf(),不用cin, cout

b)只用cin,cout,并在IO之前加入下列两条语句

- `std::ios::sync_with_stdio(false);`
- `std::cin.tie(0);`

原因：因为c++需要兼容c,所以采取了一些措施
，先把要输出的东西存入缓冲区，再输出。

- `sync_with_stdio` 这个函数是一个“是否兼容stdio”的开关，C++为了兼容C，保证程序在使用了`std::printf`和`std::cout`的时候不发生混乱，将输出流绑到了一起。而这两句话可解绑stdio
- 但是由于noip必须使用freopen,而其在stdio.h中，所以竞赛一律使用scanf/printf() (c语言读入也比c++快)

读入挂

```
• void read(int &x){
    char c = getchar();
    x = 0;
    while(c < '0' || c > '9')
        c = getchar();
    while(c <= '9' && c >= '0')
        x = x*10+c-48, c = getchar();
}
```

- void write(int x)
- {
- int num = 0;
- char c[15];
- while(x) c[++num] = (x%10)+48, x /= 10; while(num)
- putchar(c[num--]); putchar('\n');
- }

- 1. 对于每一道试题，选手只应提交一个源程序文件。源程序文件名由试题名称缩写加后缀构成，源程序文件名及后缀一律使用小写。PASCAL、C及C++程序的后缀分别为.pas, .c, 或.cpp。当参赛选手对一道试题提交多份使用不同后缀的源程序文件时，测试系统只从其中任意选取一份进行编译和评测，并以此作为选手该道试题的得分。
- 2. 使用C/C++语言者不得使用自己的头文件，使用Pascal语言者不得使用自己的库单元。除另有规定外，每道题参赛程序源文件的长度不应大于100KB，不得在规定目录下另建其它子目录。
- 3. 选手程序应正常结束并返回Linux系统，主函数的返回值必须为0。

- 4. 选手程序中只允许通过对指定文件的读写、以及对指定库函数的调用等题目中明确规定的方式与外部环境通信。在程序中严禁下列操作：
- 试图访问网络
- 使用fork、exec、system或其它线程/进程生成函数
- 打开或创建题目规定的输入/输出文件之外的其它文件和目录
- 以任何方式对标准输入/输出文件重定向
- 运行其它程序
- 改变文件系统的访问权限
- 读写文件系统的管理信息
- 使用除读写规定的输入/输出文件之外的其它系统调用
- 捕获和处理鼠标和键盘的输入消息
- 读写计算机的输入/输出端口

- 5. 现在NOIP常见的内存使用限制是256M

文件操作

- 如前所述，NOIP的题目测评本质上就是将选手提交的可执行文件，在机器上运行，要求该可执行文件可以从指定的文件读入数据，并将结果输出到指定位置。
- 因此需要理解如何在源程序中打开、读写和关闭输入数据文件和输出数据文件

文件操作

在main函数最前面加上：

```
freopen("math.in","r",stdin);
```

```
freopen("math.out","w",stdout);
```

万能头文件

- `#include <bits/stdc++.h>`



先 一 步 赢 在 未 来