

在线评测的输入输出

题目的输入数据和输出数据一般有多组（不定），并且格式多种多样。所以，如何正确处理题目的输入输出是对大家的一项基本要求，也是困扰初学者的一个难题。

基本输入

1. 输入不说明有多少个Input Block，以文件结束符EOF为结束标志。

C语法：

```
while (scanf("%d %d", &a, &b) != EOF)
{
    ....
}
```

说明：

- scanf函数返回值就是其读出的变量个数，如：

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

如果只有一个整数输入，返回值是1；如果有两个整数输入，返回值是2；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。

2. 输入的第一行提示有N个Input Block，下面接着是N个Input Block。

C语法：

```
scanf("%d", &n) ;
while (n-->0)
{
    scanf("%d %d", &a, &b);
    ....
}
```

3. 输入不说明有多少个Input Block，但以某个特殊输入为结束标志。

C语法：

```
while (scanf("%d", &n) && n!=0 )
{
    ....
}
```

说明：

- 最后一行是0。
4. 输入是一整行的字符串。

C语法：

```
char buf[BUFFER];
while (gets(buf) != NULL)
{
    ....
};
```

说明：

- 适用于字符串之间用换行符作分隔，空格可作为普通字符输入。
- 如果使用scanf，如scanf("%s%s",str1,str2)，则多个字符串之间是用一个或多个空格分隔，也可以用换行符或者TAB分隔。通常情况下，处理短字符串用scanf，处理长字符串用gets。
- 如果使用getchar函数，每次只处理一个字符，这时要注意处理读入的换行符。

5. 关于C++中cin的使用 cin在读字符串时，遇到空白符（空格，换行等）结束

```
char str[BUFFER];
while (cin >> str)
{
    ....
}
```

getline读字符串时遇到换行符结束，适用于读一整行

```
char str[BUFFER];
while (cin.getline(str, BUFFER))
{
    ....
}
```

或者用C++的string对象

```
string str;
while (getline(cin, str))
{
    ....
}
```

说明：

- cin/cout要比scanf/printf慢一些，尽可能使用scanf/printf以避免测试大量数据时因为输入输出慢而导致TLE。
- getchar/putchar要比scanf/printf更快。

基本输出

初学者一般有个误区：如果题目包含多组测试数据，他们就会把输入的内容全部保存起来，然后再依次处理。其实程序的输入和输出是相互独立的，因此，每当处理完一组测试数据，就应当按题目要求进行相应的输出，而不必将所有结果储存起来一起输出。比如：

```
#include <stdio.h>
int main(void)
{
    int a, b;
    while (scanf("%d %d", &a, &b) != EOF)
    {
        printf("a+b=%d\n", a + b);
    }
    return 0;
}
```

注意：千万不要把cout和printf混用，因为cout是带缓冲的而printf不带，所以会使得输出的数据顺序混乱。

使用文件输入输出方便调试

把下面两块宏语句分别嵌在main函数的开始和结束，这样在本地调试的时候，cin/cout和scanf/printf直接对应到指定的文件流。当提交到OJ时，这两句不会被编译，所以仍为标准I/O流，因此不用在提交前改代码。

前面宏定义里的"in.txt"和"out.txt"，根据自己的输入文件名更改即可。

```
#include <iostream>
#include <cstdio>

using namespace std;

#ifdef ONLINE_JUDGE
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#endif

#your code is here

#ifdef ONLINE_JUDGE
    fclose(stdin);
    fclose(stdout);
#endif
```

用这种方法，cin/cout和scanf/printf都可以转化为文件流。