

# 问题

不知各位大佬有没有在牛客网的笔试当中被怎么处理输入的事情烦恼过，我是每次都要纠结多长时间，浪费了宝贵的短暂笔试时间，导致最后笔试结果不理想（搞得好像搞定输入后就能写出来一样哈哈哈哈哈）。为了以后的笔试能够专心在思考算法上面，这里专门整理一下牛客网可能出现的各种输入样例的解决方案。

## C++中跟输入有关的知识

### cin>>

(1) 获取输入的一个字符或数字：`cin>>`会自动过滤掉不可见字符（如空格、回车、tab等）。若想过滤掉空字符，可以使用 `noskipws` 流进行控制。如下程序，输入的空格字符将会被存入 `input[1]` 中，也可以用于输出。

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     char input[5];
5     for(int i=0; i<5; i++) {
6         cin>>noskipws>>input[i];    // 输入： d kjh
7     }
8     for(int i=0; i<5; i++){
9         cout<<input[i];    // 输入：d kjh
10    }
11    return 0;
12 }
```

(2) 获取输入的字符串，可以用数组或 string 类型。如：

【注意】：遇到空格，回车等会结束获取输入的字符串，后面的字符串会被过滤掉（存放输入流中）。如果后面还需要输入字符串中，则会从前面存放的字符串开始获取。

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     char input[5];
5     cin>>input;    // 输入： dede po
6     cout<<input<<endl;    // 输出： dede
7     string s;
8     cin>>s;    // 这里直接从输入流中读取，不再需要我们输入
9     cout<<s<<endl;    // 输出： po
10    return 0;
11 }
```

### cin.get()

(1) `cin.get(字符变量符)`，用来接收字符，只获取一个字符，可以接收空格，遇到回车便结束。

```

1  #include <iostream>
2  using namespace std;
3  int main(){
4      char input[5];
5      for(int i=0; i<5; i++){
6          cin.get(input[i]); // 输入： f hkjil
7          cout<<input[i];   // 输出： f hkj
8      }
9      return 0;
10 }

```

(2) `cin.get(数组名, 接收字符数目)`，用来接收字符串，可以接收空格，遇回车结束。

注意：数组的最后一个字符会是'\0'，设接收字符数目为n，如果输入的字符串大于等于n，则实际接收到的输入是字符串的前面n-1个字符，包括空格（不包括回车，遇到回车就结束了），会自动在后面增加一个'\0'。

```

1  #include <iostream>
2  using namespace std;
3  int main(){
4      char input[5];
5      cin.get(input, 5); // 输入： d hlyoj
6      cout<<input<<endl; // 输出： d hl
7      return 0;
8  }

```

(3) 可以使用 `cin.get()` 来舍弃输入流中不需要的字符，或者判断输入的字符是不是换行符 `'\n'`，如果是的话可以使用 `break` 终止输入，如下的第一种输入样例。

## cin.getline()

实际是 `cin.getline(接收字符串到m, 接收个数n, 结束字符)`。接收一个字符串，可以接收空格等，最后一个字符为'\0'。结束符可以通过设置第三个参数自己设置，默认是回车。**m不能为string类型。**

注意：实际接收到的要比n少一个，因为最后一个字符为'\0'。

```

1  #include <iostream>
2  using namespace std;
3  int main(){
4      char input[5];
5      cin.getline(input, 5); // 输入： d hlyoj
6      cout<<input<<endl;   // 输出： d hl
7      return 0;
8  }

```

## getline()

用于string类的，需包含#include。使用需包含头文件#include。 `getline(cin, string s)`，接收一个字符串，可以接收空格等，不设置的话默认遇到回车停止读入操作。`getline(cin, string s)` 处理后还留有结束符在输入流中，故需要使用 `cin.get()` 来接收最后一个结束符。

**与cin.getline()的区别：**

1. `cin.getline()`接收输入字符串的是数组，`getline ()` 是string类型。
2. `cin.getline()`会在数组结尾是'\0'，`getline()`不会

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main(){
5      string s;
6      getline(cin, s); // 输入： d hlyoj
7      cout<<s<<endl; // 输出： d hlyoj
8      return 0;
9  }

```

## gets()

gets(m)用于string类的，需包含#include。可以接收空格，遇回车结束。作用跟getline()基本就是一样了。

## 可能遇到的输入样例

1、全部数据在一行中输入，没有说明输入的数字个数是多少，而且数字之间的分隔符可能是空格或者逗号。

```

1  #include <iostream>
2  using namespace std;
3  int main(){
4      vector<int> nums; // 将输入的数据先全部放到一个数组当中，之后再根据具体情况切割
      处理
5      int a;
6      while(cin>>a){ // 输入： 1,2,3,4,5,6 \n 或者 1 2 3 4 5 6 \n
7          nums.push_back(a); // 读取结束后： nums=[1, 2, 3, 4, 5, 6]
8          if(cin.get() != '\n') continue;
9          else {
10             // 在这里写具体的算法，这样便可以达到循环输入输出的问题，因为有的时候测试数
              据是多组，需要提交的代码需要做循环处理。
11         }
12     }
13     return 0;
14 }

```

2、输入的是带空白的字符串，这种情况比较可恶，因为scanf和cin都是读到空白就结束的，这时候就得使用上面知识点讲到的getline(cin, string s)来读取了。

3、预先不知道输入数据的组数——读到文件结尾

```

1  // c语言这样写
2  int a,b;
3  while(scanf("%d%d, %a, %b") != EOF){
4      printf("%d\n",a+b);
5  }
6
7  // c++这样写
8  int a,b;
9  while(cin>>a>>b){
10     cout<<a+b<<endl;
11 }

```

3、预先知道数据的组数——先读数据组数，然后循环

```

1 // c语言这样写
2 int n;
3 scanf("%d", &n);
4 for(int i=0; i<n; i++){
5     int a, b;
6     scanf("%d%d", &a, &b);
7     printf("%d\n", a+b);
8 }
9
10 // c++类似
11 int n;
12 cin>>n;
13 for(int i=0; i<n; i++){
14     int a,b;
15     cin>>a>>b;
16     cout<<a+b<<endl;
17 }

```

#### 4、只有一组数据——直接读入即可

```

1 // c语言这样写
2 int a,b;
3 scanf("%d%d", &a, &b);
4 printf("%d\n", a+b);
5
6 // C++这样写
7 int a,b;
8 cin>>a>>b;
9 cout<<a+b<<endl;

```

## 可能遇到的输出样例

- 1、不需要输出 case 数，跟上面一样，读一组，处理后输出一组即可
- 2、需要输出 case 数：

```

1 // c语言这样写
2 int n;
3 scanf("%d", &n);
4 for(int i=0; i<n; i++){
5     int a,b;
6     scanf("%d%d", &a, &b);
7     printf("Case %d %d\n", i+1, a+b); // 这里还要注意输出要求中的 Case 首字母是
//printf("Case %d %d\n\n", i+1, a+b); // 情况三
8
9 }
10
11 // c++语言这样写
12 int n;
13 cin>>n;
14 for(int i=0; i<n; i++){
15     int a,b;
16     cin>>a>>b;
17     cout<<"Case " << i+1 << a+b << endl;
18     //cout<<"Case " << i+1 << a+b << endl << endl; // 情况三
19 }

```

3、每个 case 之后有空行，这个就在上面的基础上多加一个 '\n' 即可，如上述代码注释掉的部分。

4、两个 case 之间有空行。这个的意思是最后一组输出的后面是没有空行的。

```
1 int n;
2 cin>>n;
3 for(int i=0; i<n; i++){
4     int a,b;
5     cin>>a>>b;
6     if(i > 0){
7         cout<<endl;
8     }
9     cout<<"Case " << i+1 << a+b << endl;
10    //cout<<"Case " << i+1 << a+b << endl << endl; // 情况三
11 }
```

## 一些经验与小技巧

1. 关于cin cout 和 scanf printf。做题的时候尽量使用scanf printf。下面告诉一个小常识，不要惊讶：在数据量比较大的情况下cin cout比scanf printf慢挺多。一旦遇到大数据量，光是读入就有可能跪掉。你或许可以使用std::ios::sync\_with\_stdio(false); 这条语句关掉scanf和cin的同步，加快效率。但是即使这样cin还要慢，而且一旦使用了这条语句，scanf和cin混用可能就会造成一些奇怪的错误。
2. 如果测试数据是多组的，但是恰巧你代码里面需要些标记数组，map，set等，在循环内一定记得清空，不然可能会产生前面的测试样例影响了后续数据的答案。  
(对于这个，我还没具体在C++中遇到过，但是在写python程序的时候，定义了一个字典 dict()，因为这个问题被坑了好久)
3. 不用保存所有的输入，读一组计算一组，输出一组即可

## 参考资料

[C++获取字符cin,getchar,get,getline的区别](#)  
[\(经验贴\)那些年在编程题中踩过的坑](#)  
[牛客网在线判题系统使用帮助](#)  
[OnlineJudge-OJ输入输出基础必练0813新增三题](#)