**//==================== C++ ========================**

**解释1.**

**#include <bits/stdc++.h>**

**这个头文件包含以下等等C++中包含的所有头文件：**

**#include <iostream>   
#include <cstdio>   
#include <fstream>   
#include <algorithm>   
#include <cmath>   
#include <deque>   
#include <vector>   
#include <queue>   
#include <string>   
#include <cstring>   
#include <map>   
#include <stack>   
#include <set>**

**可以查资料理解其中的类、函数**

**解释2.**

**using namespace std;**

**所谓namespace，是指标识符的各种可见范围。  
C＋＋标准程序库中的所有标识符都被定义于一个名为std的namespace中。**

**解释3.**

**ios::sync\_with\_stdio(0);**

**在c++中之所以cin，cout效率低，是因为先把要输出的信息先存入缓冲区，再输出，导致效率降低，而这段语句可以来打消iostream的输入和输出缓存，可节省时间，使效率与scanf与printf相差无几，还有应注意的是scanf与printf使用的头文件应是stdio.h而不是 iostream。**

**解释4.**

**#include <queue>**

**在C++中只要即可使用队列类，其中的成员函数如下**

**1. push()**

**2. pop()**

**3. size()**

**4. empty()**

**5. front()**

**6. back()**

**解释5. 函数next\_permutation**

**函数next\_permutation()是按照字典序产生排列的，并且是从数组中当前的字典序开始依次增大直至到最大字典序**

**#include <stdio.h>**

**#include <algorithm>**

**using namespace std;**

**int main()**

**{ int n;**

**while(scanf("%d",&n)&&n)**

**{ int a[1000];**

**for(int i=0;i<n;i++)**

**scanf("%d",&a[i]);**

**sort(a, a+n);**

**do{**

**for(int i=0;i<n;i++)**

**printf("%d ", a[i]);**

**printf("\n");**

**} while(next\_permutation(a, a+n));**

**}**

**return 0;**

**}**

**解释6. 集合函数并操作set\_union**

**#include <set>**

**set <int> A;**

**set <int> B;**

**set <int> C;**

**set <int>::iterator pos //**定义迭代器，作用是输出set元素

**cin>>m;**

**while(m--)**

**{ cin>>x;**

1. **insert(x);**

**}**

**cin>>n;**

**while(n--)**

**{ cin>>x;**

**B.insert(x);**

**}**

**cout<< “A= { ”; //输出A集合**

**for( pos=A.begin; pos!=A.end; pos++)**

**{ if( pos!=A.begin) cout<< “, ”;**

**cout <<\*pos;**

**}**

**cout<< “ } ”<<endl;**

**set\_union(A.begin, A.end, B.begin, B.end, inserter(C, C.begin()));**

**//并操作**

**cout<< “A U B= { ”; //输出并集合C**

**for(pos=C.begin; pos!=C.end; pos++)**

**{ if( pos!=C.begin) cout<< “, ”;**

**cout<< \*pos;**

**}**

**cout<< “ } ”<<endl;**

**其它使用方法查资料**

**解释7. string用法**

**String是C++标准库的一个重要的部分，主要用于字符串处理。可以使用输入输出流方式直接进行string操作，也可以通过文件等手段进行string操作。同时，C++的算法库对string类也有着很好的支持，并且string类还和c语言的字符串之间有着良好的接口。**

**#include <string>**

**string s1 = "Hello World"; int len;**

**len=s1.length(); //求串的长度**

**cout << "String1 is: " << s1 << endl;**

**const char\* s2 = s1.c\_str(); // s1 转换为 char\***

**printf("String2 is: %s \n", s2);**

**string str = “Mississippi”;  
int index;  
index = str.find\_first\_of(‘s’,0); // index为 2  
index = str.find\_first\_of(‘s’,7); // index为 -1**

**index = str.find\_last\_of(‘s’,7); // index为 6**

**string s1 = “AAA”;  
s1 += "END"; // s1 是 “AAAEND”  
s1.insert(3, “----”); // s1 是 “AAA----END”  
s1.erase(3,4); // s1 是 “AAAEND”**

**其它使用方法查资料**

**解释7. map …**

map翻译为映射，也是常见的STL容器

**map<string, int> M1;**

**M1[string(“aaa")] = 1; //直接赋值法**

**M1[string("bbb")] = 2;**

**M.insert({ string("ccc"), 5 });**

**auto p=M1.begin;**

**while(pos!=M1.end)**

**{ cout<< p->first<<”----”<<p->second<<endl;**

**p++;**

**} 其它使用方法查资料**