```cpp
1: #include<iostream>
2: #include<fstream>
3: using namespace std;
4:
5:
6: class Student{
7: public:
8:     Student(void){}
9:     ~Student();//dtor
10:     void initial(string name_,int numClass_);
11:     void add_class(int i,string className);     //store data t
12:     bool is_in(string className);
13:     string get_name();
14:
15: private:
16:     string name;
17:     int numClass;
18:     string *classList;
19: };
20:
21:
22: int main()
23: {
24:     ifstream inFile;
25:     ofstream outFile;
26:     Student *studentList;
27:     string *classList;
28:     string in_string;
29:     int numStudent,numClass;
30:
31:     //global data
32:     inFile.open("input3.txt");
33:     outFile.open("output3.txt");
34:
35:     inFile>>numStudent;
36:     studentList=new Student[numStudent];
37:     inFile>>numClass;
38:     classList=new string[numClass];
39:
40:     //import class name
41:     for(int i=0;i<numClass;i++){
42:         inFile>>in_string;
```

```cpp
43:            classList[i]=in_string;
44:        }
45:
46:        //import student data
47:        for(int i=0;i<numStudent;i++){
48:            string name;
49:            int num;
50:            inFile>>name>>num;
51:            studentList[i].initial(name,num);
52:
53:            for(int j=0;j<num;j++){
54:                inFile>>name;
55:                studentList[i].add_class(j,name);
56:            }
57:        }
58:
59:        for(int i=0;i<numClass;i++){
60:            outFile<<"Class: "<<classList[i]<<endl;
61:            outFile<<"\t";
62:            for(int j=0;j<numStudent;j++){
63:                if(studentList[j].is_in(classList[i]))
64:                    outFile<<studentList[j].get_name()<<" ";
65:            }
66:            outFile<<endl;
67:        }
68:
69:
70:
71:        return 0;
72: }
73:
74: Student::~Student()
75: {
76:     delete[] classList;
77: }
78:
79:
80: void Student::initial(string name_,int numClass_)
81: {
82:     name=name_;
83:     numClass=numClass_;
84:     classList=new string[numClass];
```

```cpp
85: }
86:
87:
88: void Student::add_class(int i,string className)
89: {
90:     classList[i]=className;
91: }
92:
93: bool Student::is_in(string className)
94: {
95:     for(int i=0;i<numClass;i++){
96:         if(classList[i]==className)
97:             return true;
98:     }
99:     return false;
100: }
101:
102: string Student::get_name()
103: {
104:     return name;
105: }
```