duskcloud [[编辑资料]](https://www.patest.cn/users/edit) [[登出]](https://www.patest.cn/users/sign_out)

* [主页](https://www.patest.cn/)
* [题目集](https://www.patest.cn/contests)
  + [基本信息](https://www.patest.cn/contests/pat-a-practise/information)
  + [题目列表](https://www.patest.cn/contests/pat-a-practise)
  + [提交列表](https://www.patest.cn/contests/pat-a-practise/submissions?self=true)
  + [排名](https://www.patest.cn/contests/pat-a-practise/ranklist)
* [帮助](https://www.patest.cn/help)

1080. Graduate Admission (30)

时间限制

200 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

It is said that in 2013, there were about 100 graduate schools ready to proceed over 40,000 applications in Zhejiang Province. It would help a lot if you could write a program to automate the admission procedure.

Each applicant will have to provide two grades: the national entrance exam grade GE, and the interview grade GI. The final grade of an applicant is (GE + GI) / 2. The admission rules are:

* The applicants are ranked according to their final grades, and will be admitted one by one from the top of the rank list.
* If there is a tied final grade, the applicants will be ranked according to their national entrance exam grade GE. If still tied, their ranks must be the same.
* Each applicant may have K choices and the admission will be done according to his/her choices: if according to the rank list, it is one's turn to be admitted; and if the quota of one's most preferred shcool is not exceeded, then one will be admitted to this school, or one's other choices will be considered one by one in order. If one gets rejected by all of preferred schools, then this unfortunate applicant will be rejected.
* If there is a tied rank, and if the corresponding applicants are applying to the same school, then that school must admit all the applicants with the same rank, *even if its quota will be exceeded.*

**Input Specification:**

Each input file contains one test case. Each case starts with a line containing three positive integers: N (<=40,000), the total number of applicants; M (<=100), the total number of graduate schools; and K (<=5), the number of choices an applicant may have.

In the next line, separated by a space, there are M positive integers. The *i*-th integer is the quota of the *i*-th graduate school respectively.

Then N lines follow, each contains 2+K integers separated by a space. The first 2 integers are the applicant's GE and GI, respectively. The next K integers represent the preferred schools. For the sake of simplicity, we assume that the schools are numbered from 0 to M-1, and the applicants are numbered from 0 to N-1.

**Output Specification:**

For each test case you should output the admission results for all the graduate schools. The results of each school must occupy a line, which contains the applicants' numbers that school admits. The numbers must be in increasing order and be separated by a space. There must be no extra space at the end of each line. If no applicant is admitted by a school, you must output an empty line correspondingly.

**Sample Input:**

11 6 3

2 1 2 2 2 3

100 100 0 1 2

60 60 2 3 5

100 90 0 3 4

90 100 1 2 0

90 90 5 1 3

80 90 1 0 2

80 80 0 1 2

80 80 0 1 2

80 70 1 3 2

70 80 1 2 3

100 100 0 2 4

**Sample Output:**

0 10

3

5 6 7

2 8

1 4

[提交代码](https://www.patest.cn/contests/pat-a-practise/1080)

模拟排序，多种关键字多种过程，思路复杂，但是理清了的话就不会有什么样例坑

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#define ll long long

#include<stack>

using namespace std;

struct stu

{

int id;

int ES;

int IS;

int now = 0;

int rank;

int admit;

vector<int>wish;

};

struct sch

{

int lim;

vector<int>admit;

};

int comp(stu a,stu b)

{

int fin\_a = a.ES + a.IS;

int fin\_b = b.ES + b.IS;

if (fin\_a > fin\_b)

return 1;

else if (fin\_a == fin\_b)

{

if (a.ES > b.ES)

return 1;

else

return 0;

}

else

return 0;

}

vector<stu>list;

int main()

{

//input

int stu\_num, sch\_num, wish;

cin >> stu\_num >> sch\_num >> wish;

vector<sch>school(sch\_num);

for (int i = 0; i < sch\_num; i++)

scanf("%d", &school[i].lim);

for (int i = 0; i < stu\_num; i++)

{

stu temp;

temp.id = i;

scanf("%d %d", &temp.ES, &temp.IS);

for (int i = 0; i < wish; i++)

{

int t;

scanf("%d", &t);

temp.wish.push\_back(t);

}

list.push\_back(temp);

}

//sorting

sort(list.begin(), list.end(),comp);

list[0].rank = 1;

for (int i = 1; i < stu\_num; i++)

{

if (list[i].ES == list[i - 1].ES&&(list[i].ES+list[i].IS)== (list[i-1].ES + list[i-1].IS))

list[i].rank = list[i - 1].rank;

else

list[i].rank = i + 1;

}

//admitting

int it = 0;

while(it<stu\_num)

{

int forward;

for (forward = it; forward < stu\_num; forward++)

{

if (forward == stu\_num-1)

break;

if (list[forward].rank != list[forward + 1].rank)

break;

}

for (int i = it; i <= forward; i++)

{

while (list[i].now != wish)

{

if (school[list[i].wish[list[i].now]].admit.size() < school[list[i].wish[list[i].now]].lim)

{

list[i].admit = 1;

break;

}

else

list[i].now++;

}

if (list[i].now == wish)

list[i].admit = 0;

}

for (int i = it; i <= forward; i++)

{

if (list[i].admit)

school[list[i].wish[list[i].now]].admit.push\_back(list[i].id);

}

it = forward+1;

}

for (int i = 0; i < sch\_num; i++)

{

if (school[i].admit.size())

{

sort(school[i].admit.begin(), school[i].admit.end());

int num = school[i].admit.size();

printf("%d", school[i].admit[0]);

for (int j = 1; j < num; j++)

printf(" %d", school[i].admit[j]);

printf("\n");

}

else

printf("\n");

}

return 0;

}