

First Southern African Regional ACM Collegiate Programming Competition Sponsored by IBM

`www.ibm.com`

National Support from EPI-USE

`www.epiuse.co.za`

Problem 2/Green Balloon: Course Scheduling

Saturday 18 September 1999

You have decided to enroll for a degree at the university. There are a total of n (where $n \leq 32$) courses, numbered from 1 to n . You are also given four specific (upper level) courses which you must complete for your degree. Before taking a course, you must first take *at least one* of the prerequisite courses. (You need not take *all* of the prerequisites.) Each course is one semester (16 weeks) long, and every course is offered every semester. There are no public holidays. The university's rules dictate that you must be present in **all** classes. The only problem with this is that you must get a ride to the university before your first class, and home again after the last one, spending all of the intervening time at the university. To minimize the temptation to drink too much caffeine on campus, you set yourself an upper-bound of k hours on campus in total throughout your degree. To satisfy this upper-bound, you write a program to schedule some courses for you.

Your input consists of several lines of text, each consisting of natural numbers (non-negative integers). The first line contains only k . Each of the following 4 lines consists of the upper level courses (given by their number) required for your degree. The next line contains n . The following n pairs of adjacent lines (thus $2n$ lines) have the following form:

1. The times of the lectures for the course are given on the first line of the pair. All lectures are one hour long and begin on the hour. Each time is given by two numbers separated by a space: the day (Monday being 1, . . . , Friday being 5, with no weekend lectures), and the time, using the 24 hour clock. Recall that all classes must be attended.
2. On the second line of the pair, the prerequisite courses are given, each as the corresponding course number. Note that a list such as 15, 23, 5 means that *at least one*, but not necessarily all, of courses 15, 23 or 5 is required.

Your output will consist of one line for each semester, beginning with your first semester. Each line contains the course numbers for the semester in ascending order, separated by a space.

An example of the input would be

```
96
4
2
3
1
4
1 12
2 3
1 14
4 3
1 7 2 7

2 8 1 8
```

There is a blank line at the end of the input A corresponding output would be

```
3 4
1 2
```