

Assignment 4

Description 1

We plan to establish a score system for lab class, and now invite you to join its development.

It is assumed that there are 14 lab classes in a semester, each lab has 5 exercises and 1 point for one .That is to say, the total score of each lab class is 5 points.

Students are required to complete these exercises and we check and mark them in class. If students don't check the results of exercise in one lab , this lab will get 0 point, and he is considered absent from the lab class.

Each student has an SID , the form of which is 8 Numbers. For example, one SID code is "20201234". The first four digits represent the year of admission and the last four are randomly generated. Note that each student has a SID, and SID is unique.

Q1: We can randomly generate the data of the student ID and the corresponding scores of 14 lab class, Since we can't provide the original data .Now we need you to write a program to complete this function.(30 marks)

Remember the following information as you write your program:

- 1) The SID of students is generated by random, and the first four digits represent the year of admission and its values are in the range[2000,2020].
- 2) The scores of students is generated by random. While the score is 0 ,it means he is absent in this lab class.
- 3) The scores of students can only be integers between 0 and 5, including 5.

Input

The Number of students.

Output

The scores of all the students in the lab class,the print format is as follows:

```
SID1: lab1's score, lab2's score,...,lab14's score
SID2: lab1's score, lab2's score,...,lab14's score
...
SIDn: lab1's score, lab2's score,...,lab14's score
```

Hint:

We can consider using a one-dimensional array to store student's ID, and a two-dimensional array to store students' score for each lab class.

Q2: With these information, we can easily find student's ID whose absent time are equal to or greater than twice and send them an email to remind them .(10 marks)

Input

none

Output

the SID of the students whose absent time equal or exceed 2

Q3: Export this information to a file named lab_records.csv (10 marks)

Input

none

Output

A file named lab_records.csv , the format is as follows:

SID1,lab1's score, lab2's score,...,lab14's score

SID2,lab1's score, lab2's score,...,lab14's score

...

SIDn,lab1's score, lab2's score,...,lab14's score

Q4: Read lab_scores.csv, calculate the average score of each lab and the average score of the course, and print out the lab ID of the average score of the lab is less than the average score of the course.(10 marks)

Input

A file named lab_records.csv

Output

The lab ID of the average score of the lab is less than the average score of the course

Note that you can do Q1,Q2,Q3 in one program, but you have to write one function for each one.

Q5:(40 marks)

You have seen during the lecture that "switch" doesn't allow using strings. It's very frequent to have programs that control other programs that run as services (we'll talk about programs that run as services and how a program can control another program towards the end of the course). For instance, there is a famous Web server program called "apache" and when you want to control it you start another program called "apache_ctl".

This program (just like a console) displays a prompt such as "> " and then expects you to type a command, for instance "start" to start the server, "stop" to stop it, "restart" to stop it first and then start it, "reload" to make it read again a configuration file that was modified, etc. Of course there is also an "exit" command to quit apache_ctl.

You are asked to write a program that will accept the following commands: **start, stop, restart, reload,status, exit** (to quit it) and, when a command other than "exit" is recognized, will simply display "command <name here> recognized". It must also say "Invalid command" if the command isn't recognized.

You are asked to test the command in a *switch* statement, not an *if ... else if ... else if ...* structure.

For this, you'll have an array of strings containing the commands, you'll search it, and if you find the command you'll return its index in the array. The index is an integer and can be used for "switch".

For legibility, you'll associate a symbol to each index.

For instance if you have

```
char *commands[] = {"start", "stop", ... };
```

You can have

```
#define START_CMD    0
```

```
#define STOP_CMD     1
```

and use in the switch:

```
case START_CMD: ...
```