1.Write a program that reads in three points describing the vertices of a triangle and computes the midpoint triangle they define, i.e. the triangle whose vertices are the three midpoints of the previous triangle. A typical run of your program should look like this:

```
Enter triangle vertices: (0,0) (0,1) (1,0)
Midpoint triangle verts: (0,0.5)(0.5,0.5)(0.5,0)
```

The struct used is defined as follow:

```
    struct point {
        double x, y;
    };
```

2.Create a struct to hold a fraction. The struct should have an integer numerator and an integer denominator member. Declare 2 fraction variables and read them in from the user. Write a function called multiply that takes both fractions, multiplies them together, and returns the result as a decimal number. You do not need to reduce the fraction to its lowest terms. Print the result of the multiplication of the 2 fraction variables.

3.Students who take the course must achieve an overall grade of 60 or higher to receive a pass certificate, with the added benefit of receiving $50 for an overall grade in the [G, 100] range and $20 for a grade in the [60, G) range. The instructor will also place the top K students in the course "Hall of Fame". In this problem, you are asked to write a program to help the teacher list the students in the Hall of Fame and count the total number of dollars given out.

Please use struct to complete this question.

Input format.

The input gives 3 integers on the first line, N (a positive integer up to 10 000 for the total number of students), G (an integer in the range [60,100] for the voucher grade cutoff described in the question), and K (a positive integer up to 100 and not exceeding N for the lowest ranking in the Hall of Fame). Each of the next N lines gives a student's account number (a string no more than 15 digits long without spaces) and overall grade (an integer in the range [0, 100]), separated by spaces. The question ensures that there are no duplicate account numbers.

Output format.

First output the total values issued in one line. Then output the rank, account number, and grade of the students who entered the Hall of Fame in non-ascending order by total grade, separated by 1 space. Note that students with the same score enjoy a tie for ranking, and when ranking is tied, the output is in alphabetically ascending order by account number.

input:

```
10 80 5
cy@zju.edu.cn 78
cy@pat-edu.com 87
1001@qq.com 65
uh-oh@163.com 96
test@126.com 39
```

```
anyone@qq.com 87
zoe@mit.edu 80
jack@ucla.edu 88
bob@cmu.edu 80
ken@163.com  70
```

output:

```
360
1 uh-oh@163.com 96
2 jack@ucla.edu 88
3 anyone@qq.com 87
3 cy@pat-edu.com 87
5 bob@cmu.edu 80
5  zoe@mit.edu  80
```