

# Phone

## Problem Description

You are looking for an internship for your summer holiday. Hendy's Mobile Corporation (HMC) called you for an interview at their company. HMC is a renowned company with multiple breakthrough innovations in the past few years. HMC is developing a smartphone with multiple storages functionality and they are looking for programmers to be part of the project.

As part of the interview questions, you are asked to create a program to analyze how an application interacts with the multiple storages in a phone.

## Input

The first line of the input contains two integers,  $N$  ( $0 < N \leq 100$ ) and  $S$  ( $0 < S \leq 1,000,000$ ), where  $N$  is the number of storages in the phone and  $S$  is the storage size of each storage. The second line contains one integer,  $A$  ( $0 < A \leq 100$ ), where  $A$  is the number of applications to be installed. The application must be installed on the first storage first, if there is enough space, before installing it on the next storage. Each line of the next  $A$  lines contains a string and an integer that represent the file name and file size accordingly. You can assume that the file name is unique and there will be no space in the name.

## Output

The first  $A$  lines are the result of each installation done on the phone.

The next two lines will be the name and size of the smallest and the largest application installed on the phone. If there are two or more applications with the same file size, output the first application encountered.

The next  $N+1$  lines contain the remaining size of each storage, with its storage name.

### Sample Input 1

```
3 10
4
HappyBird 2
IVLE 9
JavaTycoon 3
CodeCrunch 12
```

### Sample Output 1

```
Added to storage 1
Added to storage 2
Added to storage 1
Failed to be added
HappyBird 2 KB
IVLE 9 KB
Remaining size:
Storage 1: 5 KB
Storage 2: 1 KB
Storage 3: 10 KB
```

### Explanation 1

HappyBird can be added to storage 1. The remaining free space of storage 1 is  $(10 - 2) = 8$ .

IVLE with size 9 cannot be added to storage 1 with size 8. It can be added to storage 2. The remaining free space of storage 2 is  $(10 - 9) = 1$ .

JavaTycoon can be added to storage 1. The remaining free space of storage 1 is  $(8 - 3) = 5$ .

CodeCrunch cannot be added to any of the 3 storages.

**Sample Input 2**

```
2 100
6
HappyBird 28
HANDy 75
AiVeeLE 27
JavaTycoon 12
CrunchyCode 100
MySOCKs 33
```

**Sample Output 2**

```
Added to storage 1
Added to storage 2
Added to storage 1
Added to storage 1
Failed to be added
Added to storage 1
JavaTycoon 12 KB
HANDy 75 KB
Remaining size:
Storage 1: 0 KB
Storage 2: 25 KB
```

**Explanation 2**

HappyBird can be added to storage 1. The remaining free space of storage 1 is  $(100 - 28) = 72$ .

HANDy with size 75 cannot be added to storage 1 with size 72. It can be added to storage 2. The remaining free space of storage 2 is  $(100 - 75) = 25$ .

AiVeeLE can be added to storage 1. The remaining free space of storage 1 is  $(72 - 27) = 45$ .

JavaTycoon can be added to storage 1. The remaining free space of storage 1 is  $(45 - 12) = 33$ .

CrunchyCode cannot be added to both storages 1 and 2.

MySOCKs can be added to storage 1. The remaining free space of storage 1 is  $(33 - 33) = 0$ .

**Algorithm Template**

1. How are you going to store the applications inside a storage?
2. What are the criteria for an application to be failed to be added?
3. How to get the smallest/largest application?
4. How to keep track on the remaining size of all storages?