Party Schedule

You just received another bill which you cannot pay because you lack the money. Unfortunately, this is not the first time to happen, and now you decide to investigate the cause of your constant monetary shortness. The reason is quite obvious: the lion's share of your money routinely disappears at the entrance of party localities.

You make up your mind to solve the problem where it arises, namely at the parties themselves. You introduce a limit for your party budget and try to have the most possible fun with regard to this limit.

You inquire beforehand about the entrance fee to each party and estimate how much fun you might have there. The list is readily compiled, but how do you actually pick the parties that give you the most fun and do not exceed your budget?

Write a program which finds this optimal set of parties that offer the most fun. Keep in mind that your budget need not necessarily be reached exactly. Achieve the highest possible fun level, and do not spend more money than is absolutely necessary.

Input

The first line of the input specifies your party budget and the number n of parties.

The following n lines contain two numbers each. The first number indicates the entrance fee of each party. Parties cost between 5 and 25 francs. The second number indicates the amount of fun of each party, given as an integer number ranging from 0 to 10.

The budget will not exceed 500 and there will be at most 100 parties. All numbers are separated by a single space.

There are many test cases. Input ends with 0 0.

Output

For each test case your program must output the sum of the entrance fees and the sum of all fun values of an optimal solution. Both numbers must be separated by a single space.

Example

Sample input:

```
50 10
12 3
15 8
16 9
16 6
10 2
21 9
18 4
12 4
17 8
```

```
18 9

50 10
13 8
19 10
16 8
12 9
10 2
12 8
13 5
15 5
11 7
16 2

0 0
```

Sample output:

```
49 26
48 32
```