Alfred wants to plan what to cook in the next days. He can cook various dishes. For each dish the costs of the ingredients and the benefit value is known. If a dish is cooked the second time in a row, the benefit value for the second time is 50 percent of the benefit value of first time, if it is prepared for the third or higher time in a row, the benefit value is 0. For example cooking a dish with benefit value v three times in a row leads to a total benefit value of $1.5 * v$.

Help him to build the menu which maximizes the benefit value under the constraint that his budget is not exceeded.

Input

The input consists of several test cases. Each test case begins with 3 integers in a line: The number of days k ($1 \leq k \leq 21$) Alfred wants to plan for, the number of dishes n ($1 \leq n \leq 50$) he can cook and his budget m ($0 \leq m \leq 100$).

The following n lines describe the dishes Alfred can cook. The i-th line contains two integers: the costs c ($1 \leq c \leq 50$) and the benefit value v ($1 \leq v \leq 10000$) of the i-th dish.

The end of the input is signaled by a test case with $k = n = m = 0$. You don't need to process this test case.

Output

For each output, print the maximum benefit value reachable with 1 digit after the decimal point. Then print k integers with i-th integer being the number of the dish to cook on day i. Dishes are numbered from 1 to n. Print at least one space or new line character after each integer.

If there are several possible menus reaching the maximum benefit value, select the one with minimum costs, if there are several with minimum costs, you can print any of them.

If every menu exceeds the budget, print only the benefit value of '0'.

Sample Input

```
2 1 5
3 5
3 5 20
2 5
18 6
1 1
3 3
2 3
0 0 0
```

Sample Output

```
0.0

13.0
1 5 1
```