

Huge Knap Sack

Our King has won the brutal battle and this whole land is now ours. The special thing about this land is that it has many beautiful golden statues. Our King wants to take back as much gold as possible to his palace. We have found that there are **N** types of statues and -- almost unbelievably -- that there is an unlimited number of each type of statue. Each statue of type **i** has a weight of **W[i]** units and occupies **V[i]** units of volume. Our King wants to maximize the amount of gold he carries back to his palace. We may use **S** sacks for this purpose, each of volume **Y**. All sacks are filled up independently by golden statues. However, there is a provision to stitch two sacks together, at the cost of **C** units of gold. Stitching three sacks costs $2 * C$ because it requires two stitchings, and so on. Your task is to find how much gold our King can possibly gain, i.e. the total weight of the statues brought back, minus the stitching charges.

Input

T – The number of test cases.

For each test case :

N S Y C // 1st line

Next N lines two numbers W[i] and V[i] each.

Output

One integer, the maximum gain in gold for our King.

This gain is the total amount of gold transported minus stitching charges.

Constraints :

$1 \leq S \leq 1000$

$1 \leq Y \leq 1000\ 000\ 000$

$1 \leq N \leq 1000$

$1 \leq W[i] \leq 100$; (for all i)

$1 \leq V[i] \leq 18$;

The Output will fit into a 64-Bit integer.

$1 \leq T \leq 20$

All W[i] & V[i] are guaranteed to be either prime or equal to 1.

Example

Sample Input:

```
2
2 5 3 1
1 2
5 7
2 5 3 1
1 2
7 5
```

Sample Output:

```
6
17
```