# A. FU

### Description

```
Auf Wiedersehen
I got a lot of nasty things blowing up in my head
But none of them are worth my time
You ain't even worth this rhyme
And I don't I don't give a flying
```

Seaflowery and FluffyBunny are rewriting the lyrics in a song. Seaflowery has prepared $N$ words while FluffyBunny has prepared $M$ words which they can choose from. Each word has a length $l_i$ and elegance $e_i$. The two girls wants the total length of their words to be the same. Under such circumstance, they would like the sum of elegance of their chosen words to be maximized. Please tell them the maximum sum of elegance.

Note that either girl can choose no word at all.

### Input format

The first line contains a single integer $T$, denoting the number of test cases.

For each test case, the first line contains two integers $N$ and $M$.

Then $(N + M)$ lines each contain two integers $l_i, e_i$ describing the words. The first $N$ words are from Seaflowery and the other $M$ words are from FluffyBunny.

### Output format

For each test case, print a line with an integer, representing the optimal sum of elegance.

### Samples

#### Sample Input

```
2
3 4
4 7
3 8
2 2
1 4
5 8
1 3
4 4
1 2
1000 -10000
200 3000
808 5000
```

#### Sample Output

```
30
0
```

### Limitations & Hints

For 30% of the test cases, $1 \leq N, M \leq 200, 1 \leq l_i \leq 1000$

For 100% of the test cases, $1 \leq N, M \leq 1000, 1 \leq l_i \leq 1000, -10^9 \leq e_i \leq 10^9, 1 \leq T \leq 10$

It is guaranteed that the total number of $(N + M)$ in all test cases doesn't exceed 5000.

Try every possible way to solve this problem!

# B. Happy Bussing

## Description

Seems like some happy-bussing software has been updated so that the checkpoints are now randomly generated (a checkpoint is a point in 3D space).

However, **little Z** found out that the generator is not that random. It actually picks checkpoints from a huge fixed point lists, so that it appears to be randomly generated.

As we all know, one has to get to 2 checkpoints to make the record valid. **little Z** has a hacking program that controls the generation of checkpoints. He wants to know the closest pair of checkpoints so that he can finish bussing as fast as possible. Note that the distance is defined as the **Euclid Distance**.

**little Z** finds that the point list is greater than usually. So he decides to use some random algorithm to solve this problem.

## Input format

The first line contains an integer $N$, denoting the size of the point list.

For the following $N$ lines, the $i^{th}$ line contains three integers $x_i, y_i, z_i$, representing point $i$ is located at $(x_i, y_i, z_i)$.

## Output format

Output a single integer, the **square** of distance of the closest pair.


## Input format

The first line contains an integer $N$, denoting the size of the point list.

For the following $N$ lines, the $i^{th}$ line contains three integers $x_i, y_i, z_i$, representing point $i$ is located at $(x_i, y_i, z_i)$.

## Output format

Output a single integer, the **square** of distance of the closest pair.


## Samples

### Sample Input

```
7
0 6 2
14 5 5
13 0 13
0 16 4
7 16 3
13 11 4
13 17 15
```

### Sample Output

```
38
```

## Limitations & Hints

- $1 \leq N \leq 5 \times 10^5$
- $x_i, y_i, z_i \in [0, 10^9]$