Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION – 1

You are given a matrix P of N rows and M columns. It consists of integer numbers in the range [1..100]. We define the sum of a matrix is the sum of its elements. Your task is to find a submatrix Q (of A rows and B columns) of P and a submatrix K (of C rows and D columns) of Q so that the difference between the sum of Q and the sum of K is maximal, and submatrix K is absolutely inside matrix Q (i.e no element on matrix Q's sides is also in matrix K).

Because the tests are large, we suggest a method to define matrix P:
P[i][j] = (P[i][j-1] * 71 + 17) mod 100 + 1 . ($1 \le i \le N$, $1 \le j \le M$)
With this method we only care about P[i][1].

Constraints
$1 \le N , M \le 1000$
$1 \le A \le N$
$1 \le B \le M$
$0 \le C \le A - 2$
$0 \le D \le B - 2$

Input

The first line of the input contains an integer t ($1 \le t \le 10$), equal to the number of testcases. Then descriptions of t testcases follow. The first line of the description contains 6 integer numbers N, M, A, B, C, D. Then N lines follow, line i contains one integer number P[i][1].

Output

For each test case, your program should output the maximal difference between two matrices (in a separate line).

Example

Input:
```
1
3 3 3 3 1 1
1
2
3
```
Output:
```
260
```

QUESTION – 2

It's the year 21546 AD, and due to increased population (you wouldn't believe me if I gave you the actual numbers), land has become very expensive. Because of the lack of space, Heaven and Hell were built in the same area. The area can be represented as a grid of **X** × **Y** unit squares.

Some of the squares were captured by the Devil (and thus belong to Hell) and the rest is the Almighty's property. On each square, a room has been built with transparent glass walls. However, some of the heavenly rooms are already occupied by Angels. For security purposes, rooms occupied by Angels have concrete opaque walls.

Recently many fighters were killed in a tournament. Fighting is no longer considered cruel, so all the fighters will deserve spots in heaven. However, because of the space shortage, all of them may not be able to recieve a spot in heaven. The fighters still hold a grudge against each other so a fighter cannot be placed in a room from which he can see any other fighter. A fighter can only see in the four cardinal directions (North, South, East and West). He cannot look diagonally or in any other direction.

Find the maximum number of fighters who can have a heavenly room.

Input

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

The first line of each test case consists of two integers **X** <= 300 and **Y** <= 300, separated by a single space. Next, **X** lines follow, each having **Y** letters separated by spaces. The jth letter on the i-th line is one of the following (quotes are for clarity, and do not appear in the input):

1. "H", if the room at location (i, j) is heavenly and vacant.
2. "A", if the room at location (i, j) is heavenly and is already occupied by an angel. Note that these rooms are not transparent.
3. "D", if the room at location (i, j) belongs to the Devil.

Output

A single line for each test case containing an integer denoting the maximum number of fighters that can fit in heaven.

Example

Input:

```
1
4 7
H H H H H H H
H H H H H H H
H H H H H H H
H H H H H H H
```

Output:

```
4
```

QUESTION – 3

BuggyD loves to carry his favorite die around. Perhaps you wonder why it's his favorite? Well, his die is magical and can be transformed into an N-sided unbiased die with the push of a button. Now BuggyD wants to learn more about his die, so he raises a question:

What is the expected number of throws of his die while it has **N** sides so that each number is rolled at least once?

Input

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

Each test case consists of a single line containing a single integer **N** (1 <= **N** <= 1000) - the number of sides on BuggyD's die.

Output

For each test case, print one line containing the expected number of times BuggyD needs to throw his **N**-sided die so that each number appears at least once. The expected number must be accurate to 2 decimal digits.

Example

Input:
```
2
1
12
```

Output:
```
1.00
37.24
```

QUESTION – 4

Blue Mary invents a game with toy bricks. The player has N cuboids numbered from 1 to N.

The rule of the game is discribed below:

- Choose some cuboids among the N cuboids, and divide them into M(1 <= M <= N) piles,named them $Pile_1$,$Pile_2$ ... $Pile_M$. There are at least 1 cuboid in each pile. To make the game easier, for any cuboid in $Pile_K$,its id should greater than any one in $Pile_{K+1}$ (1 <= K < M).
- For each pile of cuboids,the player will put them as a tower, and he should follow the two rules below:

- The up surface of each cuboid is touched and only touched another down surface. Luckily,to make the pile looking like a tower,the up surface of the lower cuboid should cover the down surface of the higher cuboid,i.e. the length of the lower up surface is not less than that of the higher down surface, and also to the width.
- In each pile,the lower cuboid has a less id than the higher cuboid.

Your task is to find a method,to make the sum of the height of each pile maximum.

Input

The very first line of the input contain the number t,then t cases follow.

For each case,The first line contain two integer number N and M. N(N<=100) is the total number of the cuboids, M(M<=N) is the number of the piles, separated by a single space.

Then N line follow, which are the description of the cuboids 1..N. Each line contains three integer numbers(<=1000)- the length, width and height of that cuboid,separated by spaces.

Output

For each case, the output contains only one line with a single integer number - the maximum sum.

Example

Sample Input:

```
1
4 2
10 5 5
8 7 7
2 2 2
6 6 6
```

Sample Output:

```
24
```

QUESTION – 5

There are *n* dealers in the market. Each of them has some unique goods (nobody else has the same goods). Besides, each of them wants to obtain some other goods, which exist in the market. This is rather strange, but for each kind of goods on the market there exists exactly one dealer who wants to obtain it.

To prevent fraud, only exchanges in pairs are allowed in this market. Moreover, each dealer is allowed to make at most one exchange a day. But the total number of transactions isn't limited. A transaction means that all the goods of one dealer are exchanged for all the goods of the other

participating dealer (partial transactions are not allowed).

You are to write a program which outputs the minimum number of days needed for each dealer to get the goods that he wants. Also output one of the possible variants of exchanges leading to this goal.

Input

The first line contains an integer *n* [*n* <= 5000]. In the second line exactly *n* numbers of goods are given, which the dealers require. If integer *j* appears as the *i*-th at input, then this means that goods required by dealer *i* are initially owned by dealer *j*.

Output

You must output the minimum number of days *m* which are needed to complete the transactions. In the next *m* lines you must output the way these transactions should be managed by the dealers. One line corresponds to one day. At the beginning of each line you must output the number of transactions on this day. After that output the pairs of dealers who exchange their goods on this day. Dealers in pairs are separated by '-' symbol. If there are many ways to perform the exchanges then output any of them.

Example

Input:

```
7
2 1 3 5 6 7 4
```

Output:

```
2
3 1-2 4-5 7-6
1 5-7
```