Around BD Programming Contest 2020

https://toph.co/c/around-bd-2020

Schedule

The contest will run for **3h30m0s**.

Authors

The authors of this contest are AST_TheCoder, bisnu_sarkar, iammarajul, prodip_bsmrstu, RamprosadG, Riaz_BSMRSTU, and tanimahossain.

Rules

You can use C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C11 GCC 9.2, PyPy 7.1 (2.7), PyPy 7.1 (3.6), Python 2.7, and Python 3.7 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 9 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

A. It's Not Over Until It's Over

Limits 1s, 512 MB

Addy is very sad. He lost the last Intra school programming competition. He doesn’t like losing at all. Addy felt like his life was meaningless. And there’s no point of continueing programming if he can’t win.
He went to his grandpa and sat in a sad and dull mood. His grandpa understood his situation and consoled him. He said to Addy, “No matter what, Addy, You must follow your dreams. You must be passionate and dedicated to your love for something. You love programming and you do it for yourself. Not only for the trophy. You do it because it makes you happy. And about the winning? Dan Waldschmidt once said,“Trying is winning in the moment.” because Giving up is the first defeat you face and you haven’t given up yet. So, Try Addy, as long as you are trying you haven’t lost it. So, try till the last second if you can. Cause Marilyn vos Savant said,“Being defeated is often temporary condition. Giving up is what makes it permanent”
The words of his grandpa really touched Addy. Now Addy practices programming everyday and he starts with a very basic program printing the very touching quote of Dan Waldschmidt, “Trying is winning in the moment.”.
So, Today before moving forward to any other problem write a program to print Dan Waldschmidt’s quote which Grandpa told Addy and get the booster to keep trying in contest today till the last second. And from the setter’s piece of motivation, here is a quote from Yogi berra,“It’s not over until it’s over”.

Input

there is no input

Output

Output a single line with the **quote of Dan Waldschmidt** which Grandpa told Addy.

Don’t forget to print newline `('\n')` after the sentence.

B. Jhamela, Once Again

Limits 1s, 128 MB

You are given **n** positive integers as an array. Also, you are given an integer **k**. Now, try to find a subsequence of the array that **(LCM - GCD)** of that subsequence equals to **k**. You don't have to print the subsequence, just print whether it is possible to find a subsequence with given condition or not!

Note: Here, **LCM** means Least common multiple and **GCD** means Greatest common divisor. Also subsequence means a new array which can be generated by deleting some values of initial array without changing the order.

Input

You have to answer **T** independent queries.
First line contains two integers **n** & **k**, the length of the array and the value of **(LCM - GCD)**. Second line contains **n** positive integers as an array **a**. Let, the value at $\mathbf{i^{th}}$ position is $\mathbf{a_i}$.

Subtasks

Subtask #1 (10 points)
$1 \le T \le 10$
$1 \le n \le 9$
$0 \le k \le 10^2$
$1 \le a_i \le 10^2$

Subtask #2 (30 points)
$1 \le T \le 10$
$1 \le n \le 10^3$
$0 \le k \le 10^4$
$1 \le a_i \le 10^4$

Subtask #3 (60 points)
$1 \le T \le 10$
$1 \le n \le 10^5$
$0 \le k \le 10^5$
$1 \le a_i \le 10^5$

Output

Print **“YES”**, if it is possible to find a subsequence which **(LCM - GCD)** equals to **k**. Otherwise, print **“NO”**.

Print them without quotes with a newline. And, try to use faster IO as the dataset is huge.

Samples

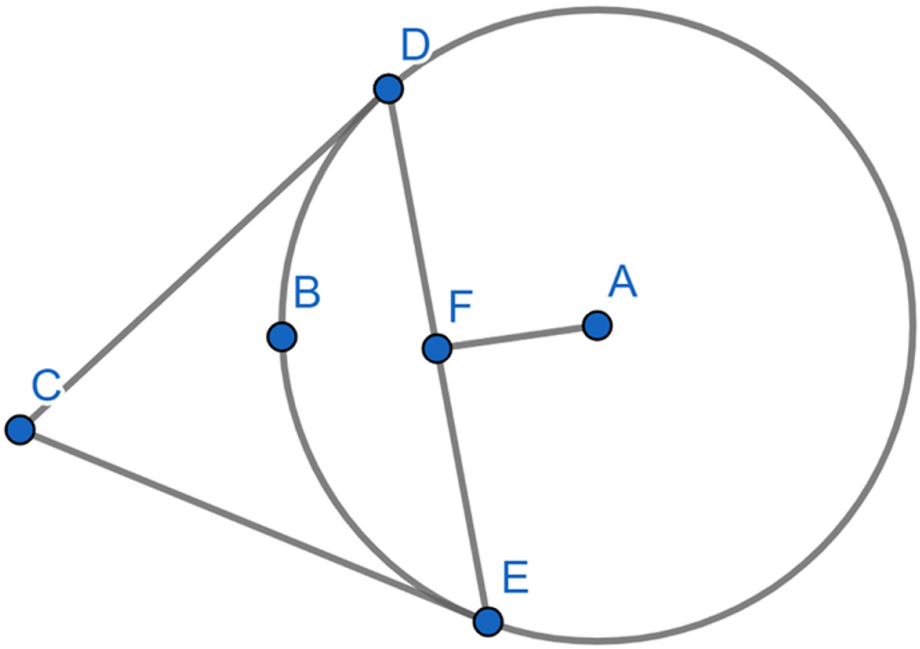
Input	Output
5 5 3 2 4 8 3 6 3 22 6 8 12 3 20 6 8 12 3 6 6 8 12 3 10 6 8 12	YES YES YES YES NO

In first case, **GCD** of **{3, 6}** is **3** and **LCM** of same subsequence is **6**. So, **(LCM - GCD)** is **3** here.
In second case, **GCD** of **{6, 8, 12}** is **2** and **LCM** of same subsequence is **24**. So, **(LCM - GCD)** is **22** here.
In third case, **GCD** of **{8, 12}** is **4** and **LCM** of same subsequence is **24**. So, **(LCM - GCD)** is **20** here.
In fourth case, **GCD** of **{6, 12}** is **6** and **LCM** of same subsequence is **12**. So, **(LCM - GCD)** is **6** here.
In last case, there’s no subsequence which difference of **LCM** and **GCD** equals to **10**.

C. Area of a Portion

Limits 500ms, 512 MB

W is working on the field from very beginning of the day. Now he has get tired and taking rest. At that time he started drawing at the ground.



Firstly he draws a circle. Then he draws two tangent from a point C(the two tangent touch the point D and E of the circle) and connects the two points with a line which are touched the circle. It created a chord named DE . Now he is thinking if he knows the shortest distance from the center to the chord AF and half length of DE, Can he find the area of CEBD portion?

After thinking a lot he find the solution. Can you also find a solution?

Input

Input start with an integer T(1<=T<=100), denoting the number of test cases.

Then T lines containing two integers x(1<=x<=10000) and y(1<=y<=10000) where x denotes the length of AF and y denotes half length of DE.

Output

For each case print the are of CEBD. Errors less than 10^{-6} will be ignored.

Samples

Input	Output
2 4 4 2 2	6.8672587713 1.7168146928

D. Nobita and Shizuka

Limits 1s, 512 MB

Probably, You all Know About The Famous Japanese Cartoon Character Nobita and Shizuka. Nobita Shizuka are very Good friend. However , Shizuka Love a special kind of string Called **Tokushuna**.

A string **T** is called **Tokushuna** if

- The length of the string is greater or equal then 3 (**|T| ≥ 3**)
- It start and end with a charecter '1' (one)
- It contain (|T|-2) number of '0' (zero)

here |T| = length of string **T** . Example , 10001 ,101,10001 is **Tokushuna** string But 1100 ,1111, 0000 is not.

One Day Shizuka Give a problem to nobita and promise to go date with him if he is able to solve this problem. Shizuka give A string **S** and told to **`Count number of Tokushuna string can be found from all possible the substring of string S`**. Nobita wants to go to date with Shizuka But You Know , he is very weak in Math and counting and always get lowest marks in Math . And In this Time Doremone is not present to help him .So he need your help to solve the problem .

Input

First line of the input there is an integer T, the number of test cases. In each test case, you are given a binary string **S** consisting only 0 and 1.

Subtasks

Subtask #1 (50 points)
1 ≤ T ≤ 100
1 ≤ |S| ≤ 100

Subtask #2 (50 points)
$1 \leq T \leq 100$
$1 \leq |S| \leq 10^5$

Output

For each test case output a line **Case X: Y** where **X** is the case number and **Y** is the number of Tokushuna string can be found from all possible the substring of string **S**

Samples

Input	Output
3 10001 10101 1001001001	Case 1: 1 Case 2: 2 Case 3: 3

Look, in first case 10001 is itself is Tokushuna string.
In second Case 2 Substring S[1-3] 101 and S[3-6] 101 Can be found which is Tokushuna string.

E. Race in DomKi

Limits 1s, 512 MB

There is a kingdom called DomKi. DomKi has N city and N-1 two-way road. Each city is numbered with **1, 2, 3, … N**. The cities of DomKi are connected to each other in such a way that it forms a Tree. Each city in DomKi is given a rank, ***r*** which is an integer.

Recently the **Tournament Planning Committee(TPC)** of the kingdom decided to arrange a **car race** in the Kingdom. In this tournament, racers from all over the world are invited to take part. So the **TPC** is a bit tense to arrange a successful tournament because finishing a tournament successfully is not an easy task.
The most important issue is the information about the tracks where the race will be held. They are so eager to know the details of the tracks before the tournament.

Note, that a track is consist of one or more roads. For better understanding, you can consider the track as the shortest route between two cities in DomKi.

Let me introduce one more thing about the track.
Suppose for a track the **TPC** has fixed two cities **a and b** where **a** is the starting city of the track and **b** is the city where the track end and called this track $\mathbf{T}_{ab}$. For a track $\mathbf{T}_{ab}$, if we store all the ranks ***r*** lying on the track $\mathbf{T}_{ab}$ and write them in a paper sequentially, then it will be an array and **TPC** called this array **R**ank **A**rray of the track **ab**, ($\mathbf{RA}_{(a,b)}$).

They, basically want to know **three** important facts about DomKi's tracks.
They listed the information, what they want to know, in a structured way like below.
* 1 a b - Is it possible to make $\mathbf{RA}_{(a,b)}$ a palindrome array by rearranging(not necessarily) the elements of the array $\mathbf{RA}_{(a,b)}$?
* 2 a b - Is the array $\mathbf{RA}_{(a,b)}$ a palindrome array ?
* 3 a b c d - Are the array $\mathbf{RA}_{(a,b)}$ and $\mathbf{RA}_{(c,d)}$ identical ?

To understand better, please see the explanation of sample case.

Meanwhile, the **TPC** have managed to know that you are a great programmer of your country, and they are planning to hire you.

Now the question is, can you help them to answer the query regarding the tracks?

Input

The input will start with an integer **T** ($T \leq 12$), denoting the number of test cases.

For each test case, In the first line, you will be given an integer **N** ($1 \leq N \leq 2 \times 10^5$), the number of cities in DomKi. The next line will contain N integers $r_1, r_2, r_3 \dots r_N$ the ranks of the cities($1 \leq r_i \leq 60$) where for every **i**($1 \leq i \leq N$), r_i is the rank of the i^{th} city.

Each of the next N - 1 lines will contain two integers **u v**, which means there is a two-way road between city u and v.

The next line will contain an integer **Q**($1 \leq Q \leq 2 \times 10^5$), the number of questions the **TPC** will ask.

Each of the next Q lines will contain a question of type **1, 2** or **3**.

It is guaranteed that there can be at most one road between any pair of cities and no road connects itself. Also, it is possible to go from each city to any other city by using some sequence of roads.

Sum of N over all test cases won't exceed 400000

Sum of Q over all test cases won't exceed 550000

Subtask Constraints

Subtask 1 (30 points)

Only queries of **type 1**.

Every city will be connected to at most two other cities directly.

$1 \leq N, Q \leq 10^3$.

$1 \leq r_i \leq 30$.

Sum of N over all test cases won't exceed 4000

Sum of Q over all test cases won't exceed 4000

Subtask 2 (70 points)

Original Constraints

Output

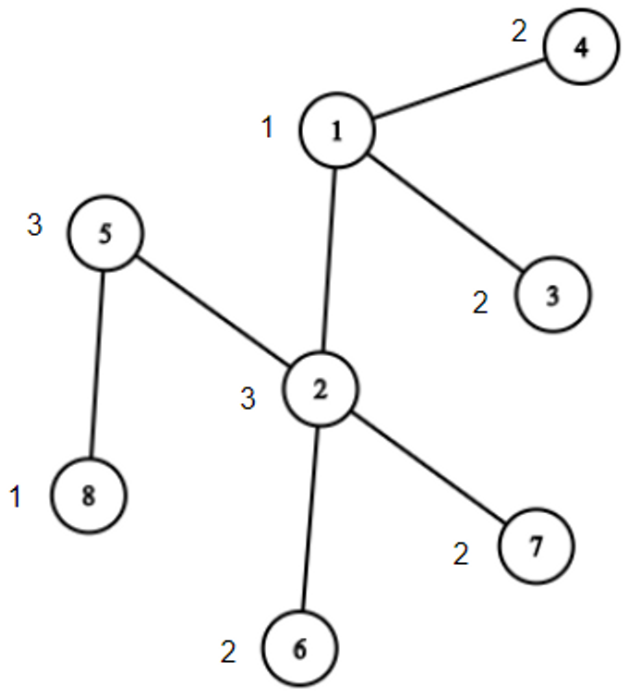
For each test case, first print the line, **"Case X:"**, where **X** is the case number and then for each query, print "YES" or "NO" (without quotes) depending on the question of the TPC, in a new line.

See the sample output for formatting.

Samples

Input	Output
1 8 1 3 2 2 3 2 2 1 8 5 6 2 5 2 3 1 1 4 1 2 2 7 6 1 5 6 1 6 8 2 1 8 2 5 7 3 3 8 4 8 3 4 5 3 7	Case 1: YES NO YES NO YES NO

The structure of the kingdom, DomKi.



For query 1, $\mathbf{RA}_{(5,6)}$ = {3, 3, 2}. By rearranging the order of the values we can make it as, $\mathbf{RA}_{(5,6)}$ = {3, 2, 3}, which is palindrome. So the answer is YES.

For query 2, $\mathbf{RA}_{(6,8)}$ = {2, 3, 3, 1}. It's not possible to make it palindrome by changing the order of the elements. So the answer is NO.

For query 3, $\mathbf{RA}_{(1,8)}$ = {1, 3, 3, 1} which is a palindrome. So the answer is YES.

For query 4, $\mathbf{RA}_{(5,7)}$ = {3, 3, 2} which is not a palindrome. So the answer is NO.

For query 5, $\mathbf{RA}_{(3,8)}$ = {2, 1, 3, 3, 1} and $\mathbf{RA}_{(4,8)}$ = {2, 1, 3, 3, 1}. Here the size of $\mathbf{RA}_{(3,8)}$ and size of $\mathbf{RA}_{(4,8)}$ are equal and for every i ($1 \le i \le$ size($\mathbf{RA}_{(3,8)}$)) $\mathbf{RA}_{(3,8)}[i] == \mathbf{RA}_{(4,8)}[i]$. So $\mathbf{RA}_{(3,8)}$ is identical to $\mathbf{RA}_{(4,8)}$.
Note, We considered both array as 1-indexed.

For query 6, $\mathbf{RA}_{(4,5)}$ = {2, 1, 3, 3} and $\mathbf{RA}_{(3,7)}$ = {2, 1, 3, 2}. Here the size of $\mathbf{RA}_{(4,5)}$ and size of $\mathbf{RA}_{(3,7)}$ are equal but for i = 4, $\mathbf{RA}_{(3,7)}[i] != \mathbf{RA}_{(4,5)}[i]$. So $\mathbf{RA}_{(4,5)}$ is not identical to $\mathbf{RA}_{(3,7)}$.

F. A Super Easy Problem - "Array Query"

Limits 1.5s, 512 MB

Let's get into the problem directly without much description this time :D .

You are given an array **A** of **N** integers. The elements of the array are $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ … $\mathbf{A}_N$. You are also given **Q** queries to perform on this array **A**. In each query, you will be given three integers **L, R, X** where **[L, R]** denotes a sub-segment of the array **A** starting at $\mathbf{A}_L$ and ending at $\mathbf{A}_R$. You have to find the minimum number of operations needed to make all values of the sub-segment, **[L, R]** of the array **A** equal to **X** i.e. $\mathbf{A}_L = \mathbf{A}_{L+1} = \mathbf{A}_{L+2} = \ldots = \mathbf{A}_R = \mathbf{X}$.

There are two types of operations. You can apply any of the two operations any number of times.

Two types of operations are described below-

Operation 1: you can choose any index i **(1 ≤** i **≤ N)** of the array **A** and then perform $\mathbf{A}_i = \mathbf{A}_i + \mathbf{1}$.
Operation 2: you can choose any index i **(1 ≤** i **≤ N)** of the array **A** and then perform $\mathbf{A}_i = \mathbf{A}_i - \mathbf{1}$.

Note: Every query is independent.

Input

First line contains an integer **T(1 ≤ T ≤ 5)**, the number of test case.
Every test case starts with an integer **N(1 ≤ N ≤ 10^5)**, the number of elements of the array **A**.
Next line will contain the elements of the array $\mathbf{A}_1$, $\mathbf{A}_2$, …, $\mathbf{A}_N$ **($-10^9 \le A_i \le 10^9$)**.
Next line will contain an integer **Q(1 ≤ Q ≤ 10^5)**, the number of queries.
Next Q lines will contain three integers **L, R(1 ≤ L ≤ R ≤ N)** and **X($-10^{12} \le X \le 10^{12}$)**.

Subtasks

Subtask #1 (10 points)
$1 \leq N, Q \leq 10^3$.

Subtask #2 (30 points)
$1 \leq N, Q \leq 10^5$.
There are at most **10** distinct values in the array **A**.

Subtask #3 (60 points)
Original Constraints.

Output

For each case, print the case number in a single line.
Then for every query print the minimum number of operations needed to make all values of the sub-segment, **[L, R]** of the array **A** equal to **X**.

Samples

Input	Output
1 3 4 5 6 2 1 3 6 1 3 4	Case 1: 3 3

In the **1^{st}** query, you have to apply **"operation 1"** in the **1^{st}** index **two times** and **2^{nd}** index **one time**. So, the answer is **3**.

In the **2^{nd}** query, you have to apply **"operation 2"** in the **2^{nd}** index **one time** and **3^{rd}** index **two times**. So, the answer is **3**.

G1. Mr. Hikiko and a Strange Game (Easy)

Limits 1s, 512 MB

Easy Version: 40 Points

The only difference between easy and hard versions is the constraint on *n* and *m*.

Mr. Hikiko recently downloaded a strange game on his phone from a store. This game is quite interesting, but he is too lazy to play a game level by level. So, he decided to write a code, which will complete all levels of this game. Now, let me tell you about his laziness a bit more. He doesn't even want to write this code himself. So, he is looking for help. Can you help him?

Rules of this game are straight forward. You will be given a grid of **n** rows and **m** columns. Each cell in this grid will have a point.

You will start from row **1**. You have to select a cell. Then you have to select **k** consecutive cells of this row including the selected cell and collect points of these cells. Then you have to make a jump from any selected cell of this row to the any cell of the next row as if their distance is less or equal to **l**. Here distance means, the absolute difference of column number between these two cells. If you make a jump from $\mathbf{cell_{1,1}}$ to $\mathbf{cell_{2,4}}$ then the distance will be |1 - 4| = 3. After making a jump, you have to select **k** consecutive cells of this row including the cell you jumped into. You will collect points from these cells and have to make another jump to the next row. You will repeat this process until you reached to the $\mathbf{n_{th}}$ row.

You can start with any cell for the first row. When you are done with all the rows, what is the maximum total points you can collect?

Input

Input will contain **T** ($1 \le T \le 10$) Test cases.
Each case will start with four integer **n**, **m**, **k** ($1 \le k \le m$), **l** ($0 \le l < m$).

Next, the grid will be given. Points in each cell will be between **0** and 10^6.

Constraint

$1 \le n, m \le 10^2$.

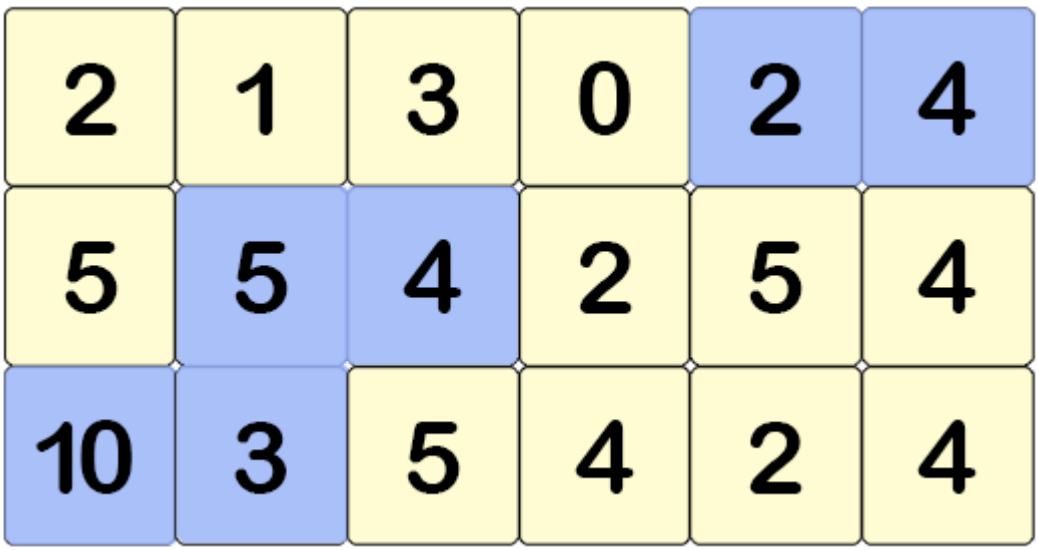
Output

For each test case, print the maximum total points you can collect.

Samples

Input	Output
1 3 6 2 2 2 1 3 0 2 4 5 5 4 2 5 4 10 3 5 4 2 4	28

Solution for the Sample Case:



G2. Mr. Hikiko and a Strange Game (Hard)

Limits 1s, 512 MB

Hard Version: 60 Points

The only difference between easy and hard versions is the constraint on *n* and *m*.

Mr. Hikiko recently downloaded a strange game on his phone from a store. This game is quite interesting, but he is too lazy to play a game level by level. So, he decided to write a code, which will complete all levels of this game. Now, let me tell you about his laziness a bit more. He doesn't even want to write this code himself. So, he is looking for help. Can you help him?

Rules of this game are straight forward. You will be given a grid of **n** rows and **m** columns. Each cell in this grid will have a point.

You will start from row **1**. You have to select a cell. Then you have to select **k** consecutive cells of this row including the selected cell and collect points of these cells. Then you have to make a jump from any selected cell of this row to the any cell of the next row as if their distance is less or equal to **l**. Here distance means, the absolute difference of column number between these two cells. If you make a jump from $\mathbf{cell_{1,1}}$ to $\mathbf{cell_{2,4}}$ then the distance will be |1 - 4| = 3. After making a jump, you have to select **k** consecutive cells of this row including the cell you jumped into. You will collect points from these cells and have to make another jump to the next row. You will repeat this process until you reached to the $\mathbf{n_{th}}$ row.

You can start with any cell for the first row. When you are done with all the rows, what is the maximum total points you can collect?

Input

Input will contain **T** ($1 \le T \le 10$) Test cases.
Each case will start with four integer **n**, **m**, **k** ($1 \le k \le m$), **l** ($0 \le l < m$).

Next, the grid will be given. Points in each cell will be between **0** and 10^6.

Constraint

$1 \le n, m \le 10^3$.

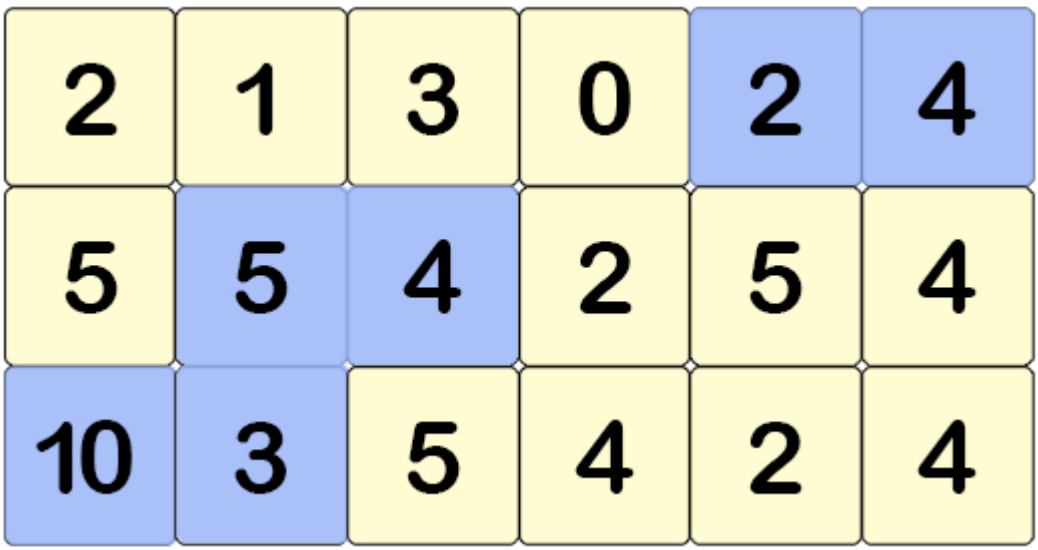
Output

For each test case, print the maximum total points you can collect.

Samples

Input	Output
1 3 6 2 2 2 1 3 0 2 4 5 5 4 2 5 4 10 3 5 4 2 4	28

Solution for the Sample Case:



H. Buckets of Water

Limits 1s, 512 MB

In Abby's area the water is available in the tap only from 6AM to 8AM. So, Abby thought she would store water during that time. But to store water Abby needs some buckets, thus, she went to the bucket section in the supermarket.
There were a lot of shelves in the bucket section. There were different sizes of buckets in different shelves. Every bucket in the same shelf has the same size. Abby saw a ladder leaning on the shelves and climbed up to reach the topmost shelf. After reaching the topmost shelf, she suddenly realized **she can hold only one bucket at a time on one hand** because she needs to use the other hand to hold on to the ladder and climb down. Note that, **she can not climb up using one hand** because she is not that strong and **once she climbs all the way down she can't climb up the ladder again** cause a lot of people are waiting to use the ladder as well.
Abby might not be strong, but she is intelligent in deed. She thought of an idea using which she can pick as many buckets as possible in such a way that she won't have to hold more than one bucket.

- She will pick a Bucket from any shelf and climb a shelf down
- if the current shelf has the larger size of buckets than Abby is holding right now, she can put the bucket she is holding into the bucket on the shelf and then pick that bucket up or She can also choose to skip the shelf if she wants.

Remember as it was said before Abby is not much strong. As the larger the buckets are the heavier. So **the total size of all the buckets she picked up can't exceed a value 'K'(this value will be given in the input)** Now, can you tell what is the maximum number of buckets Abby can pick?

Input

First line of the input there is an integer **T**, the number of test cases.
In each test case, there are two integers **N,K**, the number of shelves and the maximum total size abby can hold. In the next line there are N integers i^{th} denotes the size of buckets on i^{th} shelf. Note that the 1st shelf is closest to the ground and the N-th bucket is the top most one.

Subtasks

Subtask #1 (10 points)
1 ≤ T ≤ 10
0 ≤ K ≤200

0 < N ≤20
1 ≤ bucket size ≤ 100

Subtask #2 (30 points)
1 ≤ T ≤ 10
0 ≤ K ≤ 200
0 < N ≤ 100
1 ≤ bucket size ≤ 100

Subtask #3 (60 points)
1 ≤ T ≤ 1000
0 ≤ K ≤ 10^{18}
0 < N ≤10^{3}
1 ≤ bucket size ≤ 10^{9}
summation of all N among all test case doesn't exceed $2x10^{4}$

Output

For each testcase output a line "**Case $X: Y**" where X is the case number and Y is the maximum number of buckets picked by Abby.

Samples

Input	Output
2 9 14 6 4 2 5 7 5 3 4 1 5 9 4 3 5 2 1	Case $1: 4 Case $2: 3

In the 1st test case, Abby can pick
Bucket of size 1 from 9th shelf, skip 8th shelf and climb down
Put the Bucket of size 1 into the Bucket of size 3 and pick it up from 7th shelf, skip the 6th,5th,4th and 3rd shelves and climb down
Put the Bucket of size 3 into the Bucket of size 4 from 2nd shelf skip, pick it up, climb down
Put the Bucket of size 4 into the Bucket of size 6 from 1st shelf
Total number of buckets is 4. Total size=1+3+4+6=14 ≤ K

In the 2nd test case, Abby can pick
Bucket of size 1 from 5th shelf and climb down
Put the Bucket of size 1 into the bucket of size 2 from 4th shelf pick it up and skip 3rd shelf climb down

Put the Bucket of size 2 into the bucket of size 3 from 2nd shelf and climb down
Total number of buckets is 3. Total size=1+2+3=6 ≤ K