DIU Take-Off Programming Contest Fall 2022 Final

Daffodil International University

https://toph.co/c/diu-take-off-fall-2022-final

Toph

Schedule

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

The standings will be frozen for the last **1h0m0s** of the contest.

Authors

The authors of this contest are AK_74u, AlbinHossain, anisur_rahman, Fellow_junior, galib7112, n3wb13_223, RAIKO_, Riadh_Hasan, samiaD_R, shimul12614, and siamsplash.5.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C++20 GCC 12.1, C11 GCC 12.1, and C11 GCC 9.2 in this contest.

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

Notes

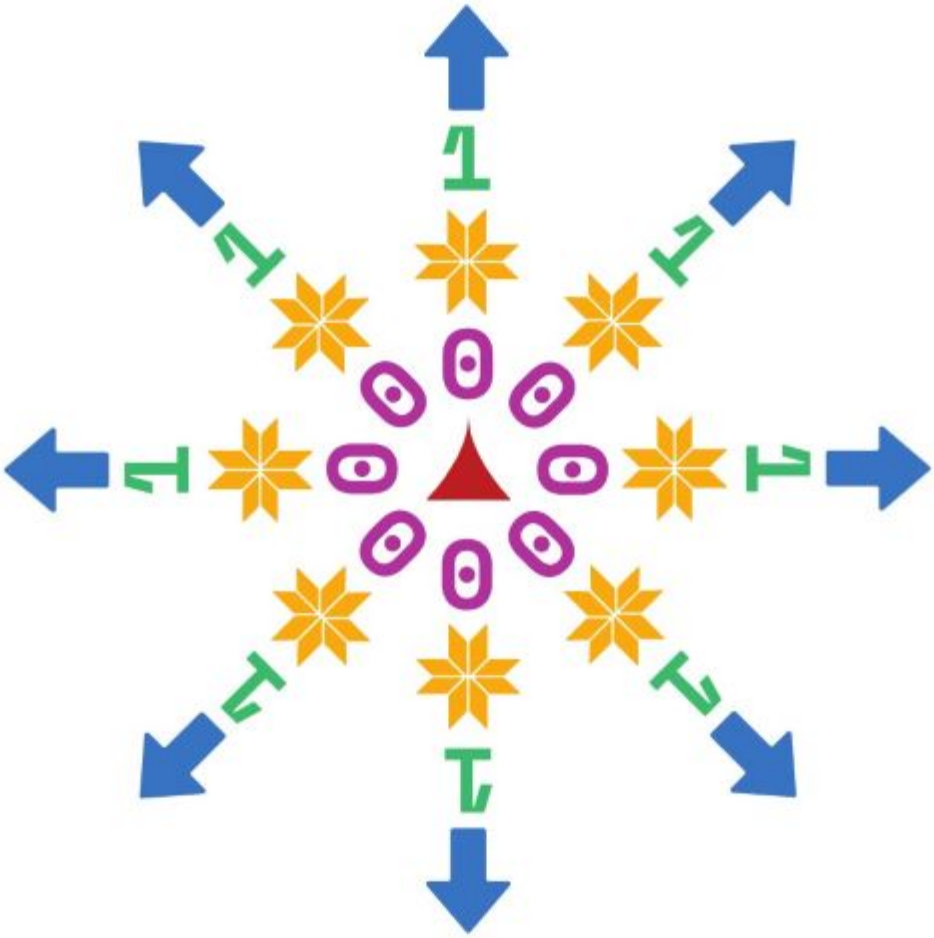
There are 11 challenges in this contest.

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. 45th ICPC Unsolved Problem!

ICPC is the oldest, largest, and most prestigious programming contest in the world. Each year, the ICPC regionals begin at local competitions among classmates to determine who may represent their university.



This year Bangladesh was the host country for **ICPC World Finals 2021**. World Finals participants are committed to advancing the art of computer science by using their skills in problem solving to solve complex algorithmic problems. Every competitive programmer has a dream to participate in this contest.

A problem was unsolved in ICPC World Finals 2021. Even the champion team **"ZEROONE"** couldn't solve it. The problem setter has put his trust on you that you can solve this problem. All you have to do is to print the name of the ICPC world Finals 2021 Champion team.

Input

There is no input for this problem.

Output

Print the the champion team name.

Be careful about the newline('\n') at the end.

B. Who Is the Champion?

The ICPC traces its roots to 1970 when the first competition was hosted by pioneers of the Alpha Chapter of the UPE Computer Science Honor Society.

The International Collegiate Programming Contest (ICPC) is the premier global programming competition conducted by and for the world's universities. Teams of three, representing their university, work to solve the most real-world problems, fostering collaboration, creativity, innovation, and the ability to perform under pressure. Through training and competition, teams challenge each other to raise the bar on the possible. It is the oldest, largest, and most prestigious programming contest in the world.

The 45th Annual ICPC World Finals is held in Dhaka, Bangladesh, hosted by the University of Asia Pacific, and organized by the Government of Bangladesh (ICT Division and Bangladesh Computer Council) for the world's future IT leaders, researchers, software engineers, and inventors. World Finals participants are committed to advancing the art of computer science by using their skills in problem-solving to solve complex algorithmic problems. It's a great honor for us to see them competing so closely.

Alice and some of her friends are **volunteers in the ICPC World Finals**. Alice is so busy distributing food among contestants. One of her friends asked- **What do you think, who will be the champion?**

You can see the scoreboard till now and guess. There are two teams where the 1^{st} team solves N problems and the 2^{nd} team solves M problems. The team which solves the maximum number of problems will be the `Champion` and the other one will be `Runner up`. As Alice is weak in logical thinking. So, she asks your help **to write a program that will determine the champion and runner up team.**

Input

The first line of input contains an integer N — denoting the number of solved problems of the 1^{st} team.

The second line of input contains an integer M — denoting the number of solved problems of the 2^{nd} team.

$1 <= N, M <= 11$

N and M are not equal.

Output

You have to output ‘‘`Champion`’’ or ‘‘`Runner up`’’ (without the quotes) for each team according to the number of solved problems. See the samples for a better understanding.

Samples

Input	Output
6 7	Runner up Champion

Input	Output
10 9	Champion Runner up

C. Opera & B2G1

"Don't practice until you get it right. Practice until you can't get it wrong"

Bangladesh is the host country for the 45^{th} Annual **ICPC World Finals 2021.**

For a successful event arrangement, a **volunteer training program** has been arranged with many trainees. The season was full of some activities like leadership, fundamental skills of volunteers, problem-solving, decision-making, and different challenges of volunteering, etc. All the trainees were very attentive and took part in the activities spontaneously.

At the end of the event, snacks will be provided for everyone. **Opera** is the organizer of the food distribution system. Currently, she has M dollars for food expenses. But she is not sure how many people will come for the snacks, so she wants to buy as many as snacks with the money she has for safety. Fortunately, a **Buy 2 Get 1 Offer** is running on pizza in a nearby shop. That means-

"If you purchase 2 pizzas then you will get 1 pizza for free".

The price of **one (1) mini-size 6-inch pizza** is X dollars.

Write a program to determine **the maximum number of pizzas she can purchase from that shop.** She can buy a pizza if and only if the total cost after purchasing that pizza does not exceed the dollar she has.

Input

The first line of the input contains T - the number of test cases. Then the following T lines contain two integers M and X separated by spaces.

$1 <= T <= 50$

$1 <= M, X <= 10^3$

Output

Output a single integer for each test case, **the maximum number of pizzas Opera can purchase.**

Samples

Input	Output
2 1000 250 515 250	6 3

D. ICPC to LCPC

In the ICPC regional contest, you have got rejected for the ICPC World Finals contest selection. But you want to participate in at least one International Contest.

Fortunately, you have got a time travel machine and have gone to the past year's N to participate in an International Contest again.

And after getting there you just discovered there is no ICPC at that time. Instead, they have an LCPC contest which is arranged only in Leap Year for a duration of 4 years. Since you wanted to participate in one International Contest, you want to participate in this LCPC world final contest.

But there is a problem. Before 1582, for a year to be called a leap year, the requirements were not like those present. It followed a rule called Julian Rule. But now everyone follows the Gregorian Rule to calculate if a year is a leap year or not.

The rules are given below:

Julian Rule: Every year that is a multiple of 4 is a leap year, i.e. has an extra day (February 29).

Gregorian Rule: The year is a multiple of 4 and not a multiple of 100 or the year is a multiple of 400.

NOTE: Julian Rule only applied before the year 1582.

Input

The input contains a single positive integer N - indicating the year you have traveled to.

$1 <= N <= 3000$

Output

You have to output a line containing "**I can participate in LCPC**" (without the quotes) **if the traveled year is a Leap Year.** Otherwise, "**I have to travel back to the past**" (without the quotes)**.**

Samples

Input	Output
1600	I can participate in LCPC

Input	Output
1581	I have to travel back to the past

E. ICPC Seat-Plan

From different countries and different universities, too many teams are coming to Bangladesh to participate in the 45^{th} Annual ICPC World Finals. The organizer team wanted to provide a suitable contest environment for the participants. So they select a big hall room where they set N rows of PCs and each row has M PCs. One team will use one PC. After completing the seat plan they have some extra seats and want to allow some extra teams.

Though everyone is busy with different types of activities, **so for the given configuration of the hall you need to calculate the number of extra teams that can participate in the contest.**

Input

The first line will take two integers N and M, which denote respectively the total row numbers and PC numbers in each row.

The next N line will take the M character "1" or "0" ("1" means the seat is booked for a team, and "0" means the seat is empty).

$(1 <= N, M <= 1000)$

Output

You have to output an integer — **the number of extra teams that can participate in the contest.**

Samples

Input	Output
5 5 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1	6

Be careful about the newline('\n') at the end.

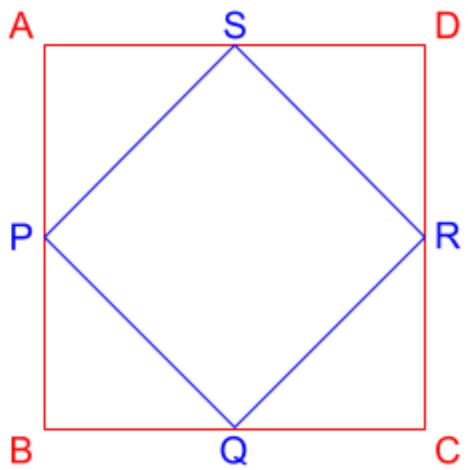
F. Squares

Three teams from your university competed in the ICPC Asia Dhaka Regional Contest 2021. Sedulous, Resolvers and Depuradors. Unfortunately, none of the teams could advance to the 45^{th} **ICPC World Finals 2021 which was hosted in Dhaka, Bangladesh**.

However, now it's time to prepare for the next ICPC World Finals. For this purpose, coach Sisir is looking for new programmers among the freshers. He has gathered some first-year students and given them some problems to solve. Whoever will be able to solve these problems, coach Sisir will consider them for team formation of the upcoming ICPC.

One of the problems is given below. Can you solve it?

$PQRS$ square is placed into $ABCD$ square such that the corners of $PQRS$ square to touch the midpoints of sides of $ABCD$ square.



Given the length of PQ. Calculate the area of $ABCD$ square.

Input

The input contains a single integer — **the length of** $PQ (1 \leq PQ \leq 10000)$.

Output

You have to output a single integer — **the area of** $ABCD$ **square.**

Samples

Input	Output
3	18

Input	Output
5	50

Input	Output
17	578

G. Kotlin Heroes

The International Collegiate Programming Contest is an algorithmic programming contest for college students. This year, the 45^{th} ICPC event was held in Dhaka. The event is held for 5-6 days. In this mega event, there are programming contests and other events like tech showcase/chilling zone, cultural experience, ICPC Challenge, and lots more fun. The Kotlin Heroes contest is one of them.
Kotlin Heroes Contest: To unveil the full power of the Kotlin programming language and their advantages in algorithmic problem-solving, this year in the ICPC World Finals event, Kotlin Heroes contest also be held. The winner will be the Kotlin Hero.

Event Details: Thus, we've invited two competitive programming stars to match their skills in solving Kotlin problems on stage at the ICPC World Finals. There will be a single problem need to be solved. **Andrew Ecnerwala**, one of the top competitive programmers in the world, and will compete in a match against **Kamil Errichto Debowski**, Codeforces "Legendary Grandmaster" and Topcoder's "Target." The match will be commentated on by none other than Roman Elizarov, Kotlin Lead Language Designer.

Problem Details: There is a race track of length **N**, in which the points are numbered from **0** to **N**. M runners will participate in a race. Every runner has power $(1 \leq i \leq m)$ P_i. Having P_i power means the i^{th} runner can only jump exactly P_i points forward. This race is quite exceptional, as there are no winner. All the runners have to meet up at **a single point** on this race-track.

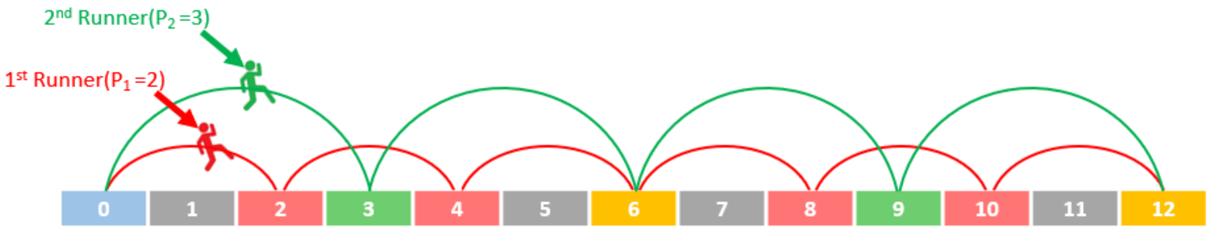
- Initially all the runners are in point 0, but this will not be regarded as a meeting point.
- A runner can either jump or wait for another runner for a meetup.
- If a runner decides to jump, then he can only jump exactly P_i step forward.

Now, the race track contains lots of points (except the initial point) such that everyone can gather and make a meetup.
You have to tell how many points are in this race track such that all runners can gather and make a meetup.

Given the value of N and M runners' power, you have to find such points where all runners can meet up.

Example:

N = 12, M = 2, P = { 2, 3 }



All the meetup points in this example are 6 and 12.

So, The number of meetup points for this race track will be 2.

Now, You have an opportunity to participate in this contest in an online mirror, the exception is that they can use only the Kotlin programming language but you can use any language.

Input

The first line of the input contains a single integer T – the number of test cases. Then the test cases follow.

- The first line of each test case contains two integers N — **the length of the race track** and M — **the number of runners**.
- The second line of each test case contains M separate integers $P_1, P_2, P_3 \dots .P_M$ — denoting the power of each runner.

$(1 \leq T \leq 100)$

$(1 \leq N \leq 10^{18})$

$(2 \leq M \leq 10)$

$(1 \leq P_i \leq 100)$

Output

For each test case output, **the total number of points where all runners can gather and make a meetup.**

Samples

Input	Output
3 12 2 2 3 100 2 2 3 20 3 1 5 10	2 16 2

H. So Long

As all people know ICPC can be very dull sometimes for people other than contestants. Jodu, Modu, and Kodu visit the contest area and watch area. They are having fun. But after some time they became bored because contestants take so much time solving exceptionally hard problems. So they come up and started playing a game.

The game goes as follows,

- **Jodu** and **Modu** will play against each other.
- There will be an empty list with **max capacity** $\mathbf{k}$.
- There will be $\mathbf{n}$ **rounds.**

In each round:
Jodu guesses a number $\mathbf{J}$
Modu guesses a number $\mathbf{M}$

if $\mathbf{M}$ is smaller than $\mathbf{J}$:

1. $\mathbf{M}$ is added to **Modu's score**.
2. Jodu will have to add the last element of the list to his score if it is smaller than $\mathbf{J}$.
3. After adding an element it must be removed.
4. Jodu will have to do this until the list is empty or he cannot add any more.
5. If the list is not full, then $\mathbf{J}$ is added to the last of the list.

else:

1. $\mathbf{J}$ is added to **Jodu's score**.
2. Modu will have to add the last element of the list to his score if it is smaller than $\mathbf{M}$.
3. After adding an element it must be removed.
4. Jodu will have to do this until the list is empty or he cannot add any more.
5. If the list is not full, then $\mathbf{M}$ is added to the last of the list.

Then the round ends.
The one having the more score will be in the lead.

Kodu have to track who is in the lead.

They have been playing for a long time. Keeping track manually Kodu got bored and started making mistakes.

Kodu has asked you to write him a program that will keep track of **each round's leading player**, given $\mathbf{J}$ and $\mathbf{M}$ of each round.

Input

The first line will be an integer $\mathbf{T}$ indicating the number of test cases. Every test case starts with Two integers $\mathbf{n}$and $\mathbf{k}$ denoting the **number of rounds** they are playing and **the maximum capacity**. The following $\mathbf{n}$ **lines** will contain two integers $\mathbf{J}$ and $\mathbf{M}$ separated by spaces.

$1 \leq T \leq 10^3$

$1 \leq n, k \leq 10^5$

$1 \leq J, M \leq 10^7$

Output

For every test case, first print a line in the format "**Case X:**" where X is the number of test cases starting from 1. Next, You have to output $\mathbf{n}$ **lines**. In the next n lines, you have to print the $\mathbf{i^{th}}$ round's leading player name "**Jodu**", "**Modu**" or "**Draw**", if both players' scores are same.

Samples

Input	Output
1 4 3 2 3 3 2 4 6 2 5	Case 1: Jodu Draw Modu Draw

Input	Output
Initially, Jodu's Score = 0 Modu's Score = 0 List = $[]$ After 1st round, ($J=2$ and $M=3$) Jodu's Score = $\mathbf{2}$ Modu's Score = 0 List = $[3]$ **Jodu** is leading After 2nd round, ($J=3$ and $M=2$) Jodu's Score = 2 Modu's Score = 2 List = $[3,3]$ It is a **draw** After 3rd round, ($J=4$ and $M=6$) Jodu's Score = 6 Modu's Score = $2 \rightarrow 5 \rightarrow \mathbf{8}$ List = $[3,3] \rightarrow [3] \rightarrow [] \rightarrow [6]$ **Modu** is leading After 4th round, ($J=2$ and $M=5$) Jodu's Score = 8 Modu's Score = 8 List = $[6,5]$ It is a **draw**	

Input	Output

`1`
`4 3`
`1000 2000`
`2048 1024`
`48 2928`
`1 100000`
	`Case 1:`
	`Jodu`
	`Jodu`
	`Modu`
	`Modu`

I. One Ball Many Chocolates

Alice, Bob and Charlie, team ABC have come to the venue of world finals to be inspired by the world finalists.

For the entertainment purpose of the contestants and the visitors, some game segments have been set up. One such game has taken the full attention of the team because of the prize. The prize is that the participant will get p chocolates if he manages to score p points in the game and all the members of team ABC love chocolates.

The game is very simple. You have to throw a ball not more than once.

Well, there are some more details.

There will be $\mathbf{n}$ blocks numbered from $\mathbf{1}$ **to** $\mathbf{n}$ and their lengths are $\mathbf{1}$ unit. You will have to stand before the first block and **throw the ball on a block whose number is at most $\mathbf{k}$**.

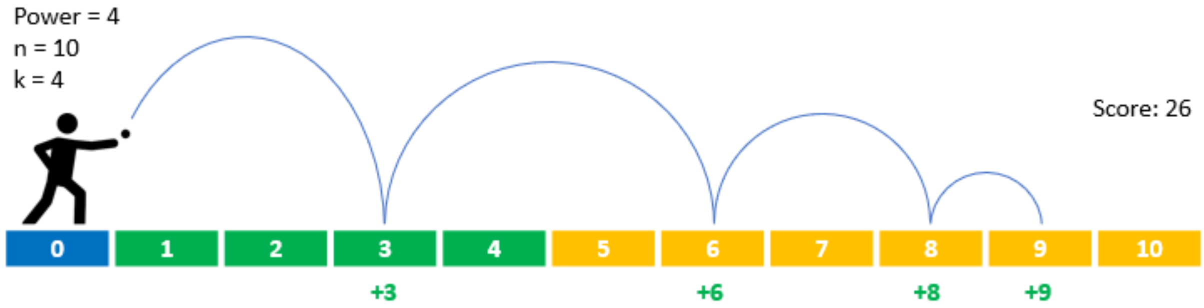
You will get $\mathbf{i}$ points if the ball lands on block $\mathbf{i}$. When the ball is thrown, you can make it bounce and land in multiple blocks to gain more points. Initially you have $\mathbf{0}$. It's possible that the ball can move out of the play area but as there aren't any blocks there no more scores would be added.

To make the ball bounce, you have to throw the ball with a certain power level. With each bounce, the ball will lose 1 power, thus will move 1 unit less than the previous bounce.

So, let's say you are playing the game where $n = 10$ and $k = 4$ you have thrown the ball with power 4 on the 3^{rd} block as it's number is less than k.

First drop will be on the 3^{rd} block with power 4.
After bouncing on 3^{rd} block, power = 3, score = 3, next block = 6^{th}
After bouncing on 6^{th} block, power = 2, score = 9, next block = 8^{th}
After bouncing on 8^{th} block, power = 1, score = 17, next block = 9^{th}
After bouncing on 9^{th} block, power = 0, score = 26, next block = having no power, none.

This is just an example. There might be a more optimal solution.



Now, you are not playing the game. But you can see Team ABC's interesting approach to gain the maximum amount of points. Alice is thinking of a clever solution to gain the most points. Bob on the other hand, started to calculate all possible outcomes for all possible actions to find the most winning strategy, But Charlie has thrown the ball and won the maximum amount of points and the maximum number of chocolates one can get from that game. The announcer specially announced this incident but didn't disclose how many points Charlie acquired. As Charlie and the team have already won a huge amount of chocolates, they stopped investing their time on this game and moved on to the next one.

You as an observant one, **want to find out the exact points Charlie earned**, as you also love chocolates. You already realized that having more power always won't do you any good, **so you will never throw the ball with power more than $\mathbf{n}$**.

Given $\mathbf{n}$ and $\mathbf{k}$, you have to find out **the maximum score(chocolates) one can get from the game**.

Input

The first and only line of input will contain 2 space separated integers $\mathbf{n}$ and $\mathbf{k}$.

$$1 \leq k \leq n \leq 10000$$

Output

You have to output an integer which is the **maximum number of chocolates you can gain from the game.**

Samples

Input	Output
`10 5`	`30`
Throwing the ball on the 4^{th} block with power 4 will yield the maximum score.	

Input	Output
28 5	126
Throwing the ball on the 5^{th} block with power 7 will yield the maximum score.	

J. Freezed Standings

Team **Depuradores** and Team **Resolvers** represented their university in the **ICPC world finals 2021**. Depuradores and Resolvers both teams solved four problems in the world final.

Depuradores was hoping to become the **Asia West Champion** as they were in the first position from Asia West before the **freezing of standings.** But in the freezing time, another Asia west team solved one more problem which increased their solve count than Depuradores. So, in the prize-giving ceremony, Depuradores was able to know that they became **Second in the Asia West Region.**

After returning back to their university both teams went to meet their coaches and other teams. Team **Sedulous** was asking them for the problem set of the World final. After getting the problem set Team Sedulous found a very interesting problem. Team Sedulous **was not that strong; they were struggling to solve that problem.** One of the members of Sedulous got to know that you have a very strong zone in problem-solving. Now he is asking for your help to solve the problem. The problem description is —

Aliban lives in an imaginary world where everyone uses **imaginary language**. That language has an **infinite number of characters.** One day **Aliban's** friend **Uban** came to her and asked her — "You are given two integers N and K. A string S is constructed using the first N distinct characters of that imaginary language. Another string T is obtained by concatenating S, K times. How many distinct substrings are there in the string T where at least one character has occurred more than once?"

Input

The single line of input contains two integers N and K (Here N represents the number of distinct characters that are used to construct string S. K represents how many times string S is concatenated to obtain string T)

$1 <= N, K <= 10^{15}$

Output

In the output, have to print a single integer X. Here X is the number of distinct substrings in the string T where **at least one character has occurred more than once**. Since the answer can be very large, **print it modulo** 998244353**.**

Samples

Input	Output
3 2	6

$N = 3$, consider the first 3 characters are a, b, c. So string $S =$ "abc" and $K = 2$. Initially, string T is empty after concatenating S two times - $T =$ "abcabc".

Now in string T**,** there are 21 substrings (size of T, $n = 6$, total substring $= \frac{n \times (n+1)}{2} = 21$. From all 21 substrings only 15 substrings are distinct - a, ab, abc, **abca**, **abcab, abcabc,** b, bc, bca, **bcab, bcabc,** c, ca, cab, **cabc**.

Out of 15 substrings, only 6 distinct substring has at least one character that appears more than once.

Input	Output
1000 1000	256147

K. A Giveaway

Mr. Tom is the head of ICPC world final organizing committee. For the world final, he comes to Dhaka. For buying some goodies he goes to a departmental store. He sees an interesting thing in this store. This store gives some gift cards on some products by following some rules.

There is n pile of offer products. Each pile contains two products with sequence order as(1-st product, 2-nd product).

Each product contains:

- A single product has a price p[i].
- Also a single product contains 2 x n gift cards collection in a row (g[1], g[2], g[3] ... g[2 x n]) where g[i] denotes the amount of money.

He has m amount of money. He wants to buy those offered products using less than or an equal m amount of money and maximize the total amount of gift card money. But this time buying a product is different.

For buying i-th product:

- He needs at least p[i] amount of money.
- From i-th pile he can buy 1-st product first. More formally (to buy 2nd product he needs to buy 1-st product first).
- During buying i-th product, he gains i-th gift card money as g[i] from this particular product's gift card collections.

Input

The first line of input contains an integer **n, m** - the number of piles and amount of money Mr tom has.

The next two line contains $2 \times n$ integers where:

- the 1-st line contains **n** integers as 1st product price as (p[1],p[2],p[3],.....p[n]) of i-th piles.
- the 2-nd line contains **n** integers as 2-nd product price as (p[1],p[2],p[3],.....p[n]) of i-th piles.

Then the next **N** line as (1,2,3,..N) i-th line where:

- each i-th line contains $2 \times n$ integers as i-th piles 1-st product gift card money as $(g[1], g[2], g[3] \ldots .g[2 \times n])$.

Then the next **N** line as (1,2,3,..N) i-th line where:

- each i-th line contains $2 \times n$ integers as i-th piles 2-st product gift card money as $(g[1], g[2], g[3] \ldots .g[2 \times n])$.

$1 \leq n \leq 14$

$1 \leq m, UC_i, BC_i, g_i \leq 10^{17}$

Output

Print the maximum total price of the gift cards.

Samples

Input	Output
2 1000 39 40 16 89 19 53 35 58 23 45 93 15 9 24 50 21 20 11 59 44	180