Problem A. Yoga Day

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Surya Namaskar, also known as Sun Salutation, is a series of 12 yoga poses performed in a sequence.

Chef repeats this sequence of yoga poses multiple times during his session.

Given that Chef has performed N yoga poses, find the number of rounds of Surya Namaskar he completed during the session.

Input Format

The input will contain a single integer N, denoting the number of yoga poses Chef performed during his session.

Output Format

Output the number of rounds of Surya Namaskar Chef completed during the session.

Constraints

- $1 \leq N \leq 100$

Sample 1

Input	Output
55	4

Chef completed 4 rounds of Surya Namaskar comprising of $4 \cdot 12 = 48$ yoga poses. The fifth round was incomplete since Chef performed only 7 poses in that round.

Sample 2

Input	Output
`11`	`0`

Chef performed 11 yoga poses whereas each Surya Namaskar consists of 12 poses. Thus, he did not complete even 1 round of Surya Namaskar.

Sample 3

Input	Output
`24`	`2`

Since Chef performed 24 yoga poses and each Surya Namaskar consists of 12 poses, he completed 2 rounds of Surya Namaskar.

Problem B. Yoga Class

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

On the occasion of Yoga Day, the only yoga instructor in Chefland, Chef, has received numerous queries.
Chef has managed to free up N hours from his busy schedule to conduct yoga sessions.
There are two types of sessions that Chef offers:

- Type 1 session, which lasts 1 hour, and earns Chef X rupees.
- Type 2 session, which lasts 2 hours, and earns Chef Y $(Y > X)$ rupees.

Find the **maximum** amount of money Chef can earn in N hours.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of three space-separated integers N, X, Y — the number of hours for which Chef will take yoga sessions, price of type 1 session, and price of type 2 session respectively.

Output Format

For each test case, output on a new line, the **maximum** amount of money Chef can earn in N hours.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq N \leq 1000$
- $1 \leq X < Y \leq 100$

Sample 1

Input	Output
3	20
4 5 7	3
3 1 2	14
5 2 6	

Test case 1: Chef will take type 1 sessions for all 4 hours. Thus, in 4 hours, he can take 4 sessions earning a total of $4 \cdot 5 = 20$ rupees.

Test case 2: Chef will take one session of type 1 and one session of type 2. Thus, in 3 hours, he earns a total of $1 \cdot 1 + 1 \cdot 2 = 3$ rupees.

Test case 3: Chef will take one session of type 1 and two sessions of type 2. Thus, in 5 hours, he earns a total of $1 \cdot 2 + 2 \cdot 6 = 14$ rupees.

Problem C. Make Permutation

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Chef has an array A of size N. He wants to make a permutation† using this array.

Find whether there exists an array B consisting of N non-negative integers, such that the array C constructed as $C_i = A_i + B_i$ is a permutation.

† A permutation of size N is an array of N distinct elements in the range $[1, N]$. For example, $[4, 2, 1, 3]$ is a permutation of size 4, while $[3, 2, 2, 1]$ and $[1, 3, 4]$ are not.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case consists of N - the size of the array A
  - The next line contains N space-separated integers - $A_1, A_2, \ldots, A_N$ - the elements of array A.

Output Format

For each test case, output `YES` if there exists an array B such that array C constructed as $C_i = A_i + B_i$ is a permutation, otherwise output `NO`.

You may print each character of the string in uppercase or lowercase (for example, the strings `YES`, `yEs`, `yes`, and `yeS` will all be treated as identical).

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $1 \leq A_i \leq N$

Sample 1

Input	Output

4
5
4 1 3 2 1
5
2 4 3 4 2
1
1
6
1 1 1 1 6 6
	YES
	NO
	YES
	NO

Test case 1 : Consider $B = [0, 4, 0, 0, 0]$. The corresponding array C becomes $[4, 5, 3, 2, 1]$, which is a permutation. Some other possible values of B for which C is a permutation are $[1, 0, 1, 1, 1]$ and $[1, 3, 0, 0, 0]$.

Test case 2 : It can be proven that no valid array B exists.

Problem D. Trick Or Treat

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Halloween is approaching, and you plan to distribute sweets to the children visiting your house. You expect a total of M children.

You have N boxes of candies and N boxes of chocolates. The i^{th} candy box contains A_i candies, and the i^{th} chocolate box contains B_i chocolates.

Your task is to choose **exactly one** box of candies and **exactly one** box of chocolates.
You will then mix the selected candies and chocolates and distribute the combined sweets equally among the M children such that no sweet is left.
Each child should receive the same number of sweets, but the types of sweets do not need to be identical.

Count the number of ways to choose 1 box of candy and 1 box of chocolates to be able to evenly distribute the sweets among the M children.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains two space-separated integers N and M — the number of boxes and the number of children.
  - The second line contains N space-separated integers $A_1, A_2, \ldots, A_N$, denoting the number of candies in each box.
  - The third line contains N space-separated integers $B_1, B_2, \ldots, B_N$, denoting the number of chocolates in each box.

Output Format

For each test case, output the number of ways of choosing 1 box of candy and 1 box of chocolates to evenly distribute the sweets among the M children.

Constraints

- $1 \leq T \leq 10^3$
- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq M \leq 100$
- $1 \leq A_i, B_i \leq 10^9$
- The sum of N does not exceed $2 \cdot 10^5$ over all test cases.

Sample 1

Input	Output
`3`	`5`
`4 5`	`9`
`2 8 3 10`	`0`
`5 12 15 13`	
`3 1`	
`1000000000 1000000000 1000000000`	
`1000000000 1000000000 1000000000`	
`1 100`	
`23`	
`76`	

Test case 1: There are 5 ways to select the boxes such that sweets can be evenly distributed: - Select candy box 1 and chocolate box 4 to have total of $2 + 13 = 15$ sweets. This way we can distribute 3 sweets each. - Select candy box 2 and chocolate box 2 to have total of $8 + 12 = 20$ sweets. This way we can distribute 4 sweets each. - Select candy box 3 and chocolate box 2 to have total of $3 + 12 = 15$ sweets. This way we can distribute 3 sweets each. - Select candy box 4 and chocolate box 1 to have total of $10 + 5 = 15$ sweets. This way we can distribute 3 sweets each. - Select candy box 4 and chocolate box 3 to have total of $10 + 15 = 25$ sweets. This way we can distribute 5 sweets each.

Test case 2: Any candy box can be chosen along with any chocolate box.

Test case 3: Since we have only one candy box and one chocolate box, choosing them would result in total of $23 + 76 = 99$ sweets which cannot be evenly distributed between 100 children.

Problem E. Split Permutation

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

The *goodness* of a permutation is the **maximum** integer k such that the permutation can be split[‡] into **exactly** k subarrays where each subarray has the **same sum**.

You are given an integer N. Generate a permutation of length N which has **maximum** *goodness*.

If multiple permutations with maximum goodness exist, print any.

[†]A permutation of length N consists of all integers from 1 to N in any order.
[‡]A split of permutation is the distribution of the elements of permutation such that each element lies inside **exactly one** subarray. For example, for a permutation $[4, 1, 5, 3, 2]$; $\{[4, 1], [5, 3], [2]\}$ is a split whereas $\{[4, 1], [3], [2]\}$ and $\{[4, 1, 2], [5, 3]\}$ are not.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of a single integer N — the length of the required permutation.

Output Format

For each test case, output on a new line, N space-separated integers, denoting a permutation of length N with maximum goodness.

If multiple permutations with maximum goodness exist, print any.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq N \leq 10^5$
- The sum of N over all test cases won't exceed 10^6.

Sample 1

Input	Output

3
1
2
3
	1
	1 2
	1 2 3

Test case 1: We can only split the permutation of length 1 into 1 subarray.

Test case 2: If we split the permutation into two subarrays, the subarray sums would be 1 and 2. Thus, we can only split the permutation into 1 subarray.

Note that both $[1, 2]$ and $[2, 1]$ are acceptable permutations.

Test case 3: Consider the permutation $[1, 2, 3]$ which can be split into $\{[1, 2], [3]\}$ where both subarrays have the sum 3.

It can be shown that any permutation of length 3 cannot be split into more than 2 subarrays while satisfying the conditions.