Problem A: Hello, world!

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

You are to write the most basic program. It should just output **"Hello, world!"** on a single line, no matter what the input.

Sample input and output

Input	**Output**
12345	Hello, world!

Problem B: A + B

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Planet Earth is under attack. The only way to save the World is to calculate the sum of two numbers. You are the only one who is able to manage it.

Input

The input contains two integers A and B ($-10^4 \leq A, B \leq 10^4$).

Output

Output a single integer - the sum of A and B.

Sample input and output

Input	**Output**
2 2	4
-10 5	-5

Problem C: Buoys

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

You are a lifeguard at the beach. One day you noticed that some of the buoys which limit the area that you are allowed to swim in go underwater at high tide. To rectify the situation, you are to determine which buoys need longer chains.

Input

The input consists of

- one line containing two integers N and f ($1 \leq N \leq 100, 0 \leq f \leq 500$) – the number of buoys at the beach and the height of the flood in cm
- one line containing N integers h_i ($0 \leq h_i \leq 1000$), the maximal height at which buoy i is able to float ($1 \leq i \leq N$).

Output

Output the indices of the buoys which need longer chains in ascending order.

Sample input and output

Input	**Output**
3 350 300 400 250	1 3

Problem D: Queue

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Yesterday the new smartphone named yPhone was presented and today excited customers are waiting all day to be the first who enter the shop that will sell these magic phones. The shop has two entrances. The queue starts at one entrance and ends at the other. Nobody knows which door will open. Notice that the queue is a straight row of people standing one after another.

Each customer has its own number. You have the list of those customer numbers in the order in which they stand in the queue. Maybe the order in which they will enter the shop is the same as in your list... but maybe not (you still don't know which door will be opened). To be sure you need to make another list in case the other door will be opened.

Input

The input starts with the number t of test cases ($1 \leq t \leq 100$). Each test case consists of

- one line containing an integer N ($1 \leq N \leq 2048$) specifying the number of customers in the queue
- one line containing the list of the customers' numbers $n_1, ..., n_N$ in the order in which they stand in the queue starting from one of the shop's doors ($1 \leq n_i \leq 10^8$).

Output

For each test case output a single line which contains the reversed list of the customers' numbers.

Sample input and output

Input	Output
3 5 4 9 8 2 1 1 3 6 1 8 9 7 4 3	1 2 8 9 4 3 3 4 7 9 8 1

Problem E: Squabbits

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Squabbits are fairly strange, magical creatures: Most of the time they wander the world all on their own, but whenever one of them discovers a field of squarrots – their favorite food – it starts to rapidly summon other squabbits. Just recently scientists were finally able to measure the exact pattern: In the first second, three more squabbits appear. In every second after that, two more squabbits appear than in the second before (so five in the second second, seven in the third and so on). The scientists would really like to describe this with a mathematical model, but they are very bad at math, so for starters they ask you to write a program, which, given a positive integer N, calculates the amount of squabbits after N seconds in the scenario described above.

Input

The input starts with the number t of test cases ($1 \leq t \leq 100$). Each test case consists of an integer N denoting an amount of seconds ($0 \leq N \leq 10^9$).

Output

For each test case output a single line which contains the number of squabbits after the given amount of seconds, as specified above.

Sample input and output

Input	Output
3 0 16 38	1 289 1521