Efficient Cashier (Easy Version)

Simon is a cashier and he is very good at giving change to customers who pay. He always makes sure that he gives them the least amount of coins needed. The coins in the cash register are your traditional coins: the quarter, dime, nickel, and penny {25, 10, 5, 1}. Conveniently, the cash register also has an unlimited amount of each of these coins. You will be given *t* customers to make change for. Each customer needs *n* cents in change. Can you figure out how many coins Simon gives to each customer?

Input

The first line of input contains *t* ($1 \leq t \leq 20$), the number of different customers he has to give change to.

For each test case you will be given *n* ($1 \leq n \leq 10^{12}$), the number representing how many cents the customer needs in change.

Note that the number of change that each customer needs can exceed the storage of an 32 bit integer!

Output

For each customer, say how many coins Simon gives them.

Input	Output
3 17 50 9	4 2 5

For the first customer, Simon will give them a dime, a nickel, and two pennies.
For the second customer, Simon will give them 2 quarters.
For the third customer, Simon will give them a nickel, and 4 pennies.

Efficient Cashier (Hard Version)

Simon is a cashier and he is very good at giving change to customers who pay. He always makes sure that he gives them the least amount of coins needed. Simon now lives in a foreign country and there are k different types of coins, each one worth ki. Conveniently, the cash register has an unlimited amount of each of these coins. You will be given *t* customers to make change for. Each customer needs *n* cents in change. Can you figure out how many coins Simon gives to each customer?

Input

The first line of input contains *t* ($1 \leq t \leq 20$), the number of different customers he has to give change to.

For each test case you will be given *n* ($1 \leq n \leq 10^5$), the number representing how many cents the customer needs in change and *k* ($1 \leq k \leq 20$), the number of different coins in the register.

On the next line you will be given *k* numbers, each one representing *ki* ($1 \leq ki \leq 1000$), how many cents this coin is worth. It is guaranteed that each coin has a different value.

Output

For each customer, say how many coins Simon gives them.
In the case that it is impossible to provide change for this customer, output -1.

Input	Output
3 14 3 1 7 10 15 3 2 16 20 12 3 2 5 15	2 -1 3

For the first customer, Simon will give the customer two 7 cent coins.
For the second customer, it is impossible for Simon to make change.

For the third customer, Simon will give them two 5 cent coins and one 2 cent coin.

Dungeon Treasure

Heidi is playing the video game Skyrim. She enters a dungeon and sees *n* pieces of loot with a certain weight and value. Heidi's backpack however only has a certain capacity *k* and she cannot leave the dungeon if the total weight of the items that she chooses exceeds the capacity of her backpack. Can you ensure that she is able to leave the dungeon with the maximum value possible? Note that each item is UNIQUE meaning she cannot take two of the same item.

Input

The first line of input contains *t* ($1 \leq t \leq 10$), the dungeons that she has to do this procedure for.

The first line of each test case contains *n* ($1 \leq n \leq 20$), the number of pieces of loot that she finds in the dungeon followed by *k* ($1 \leq k \leq 200$), the maximum capacity of her backpack.

On the next *n* lines there will be two numbers for each of the pieces of loot, *w* ($1 \leq w \leq 100$) and *v* ($1 \leq v \leq 200$) where *w* is the weight of this current item and v is the value for this item.

Output

For each dungeon output the maximum value that Heidi can retrieve.

Input	Output
3 3 10 4 5 4 5 9 8 3 15 1 100 15 500 1 100 1 1 5 250	10 500 0

In the first dungeon, Heidi should take the first and second items both worth 5 each giving a total answer of 10.
In the second dungeon, Heidi should only take the second item to get a maximum value of 500.

In the last dungeon, Heidi does not have the capacity to take anything so she leaves with 0 value.

Tyler and Line

Tyler is waiting in a long line *n* people in it (including himself). He was curious as to how many different ways everyone in line can order themselves. Can you answer this question for him? For example, if there are 3 people in line here are all of the possibilities.

1, 2, 3

1, 3, 2

2, 1, 3

2, 3, 1

3, 1, 2

3, 2, 1

Input

The first line of input contains *t* ($1 \leq t \leq 20$), the number of queries to answer.

For each query you will be given *n* ($1 \leq n \leq 10^5$), the number of people in line.

Output

For each query output the number of unique ways that the people in line can order themselves. Since the answer can be very large, output it modulo 1000000007.

Input	Output
4 1 2 3 50	1 2 6 318608048