Treino para iniciantes #13

A. Interesting drink

2 seconds, 256 megabytes

Vasiliy likes to rest after a hard work, so you may often meet him in some bar nearby. As all programmers do, he loves the famous drink "`Beecola`", which can be bought in n different shops in the city. It's known that the price of one bottle in the shop i is equal to x_i coins.

Vasiliy plans to buy his favorite drink for q consecutive days. He knows, that on the i-th day he will be able to spent m_i coins. Now, for each of the days he want to know in how many different shops he can buy a bottle of "`Beecola`".

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the number of shops in the city that sell Vasiliy's favourite drink.

The second line contains n integers x_i ($1 \le x_i \le 100\,000$) — prices of the bottles of the drink in the i-th shop.

The third line contains a single integer q ($1 \le q \le 100\,000$) — the number of days Vasiliy plans to buy the drink.

Then follow q lines each containing one integer m_i ($1 \le m_i \le 10^9$) — the number of coins Vasiliy can spent on the i-th day.

Output

Print q integers. The i-th of them should be equal to the number of shops where Vasiliy will be able to buy a bottle of the drink on the i-th day.

input

```
5
3 10 8 6 11
4
1
10
3
11
```

output

```
0
4
1
5
```

On the first day, Vasiliy won't be able to buy a drink in any of the shops.

On the second day, Vasiliy can buy a drink in the shops 1, 2, 3 and 4.

On the third day, Vasiliy can buy a drink only in the shop number 1.

Finally, on the last day Vasiliy can buy a drink in any shop.

B. Candies

1 second, 256 megabytes

After passing a test, Vasya got himself a box of n candies. He decided to eat an equal amount of candies each morning until there are no more candies. However, Petya also noticed the box and decided to get some candies for himself.

This means the process of eating candies is the following: in the beginning Vasya chooses a single integer k, same for all days. After that, in the morning he eats k candies from the box (if there are less than k candies in the box, he eats them all), then in the evening Petya eats 10% of the candies **remaining** in the box. If there are still candies left in the box, the process repeats — next day Vasya eats k candies again, and Petya — 10% of the candies left in a box, and so on.

If the amount of candies in the box is not divisible by 10, Petya rounds the amount he takes from the box down. For example, if there were 97 candies in the box, Petya would eat only 9 of them. In particular, if there are less than 10 candies in a box, Petya won't eat any at all.

Your task is to find out the minimal amount of k that can be chosen by Vasya so that he would eat at least half of the n candies he initially got. Note that the number k must be integer.

Input

The first line contains a single integer n ($1 \leq n \leq 10^{18}$) — the initial amount of candies in the box.

Output

Output a single integer — the minimal amount of k that would allow Vasya to eat at least half of candies he got.

input

```
68
```

output

```
3
```

In the sample, the amount of candies, with $k=3$, would change in the following way (Vasya eats first):

$$68 \to 65 \to 59 \to 56 \to 51 \to 48 \to 44 \to 41 \\ \to 37 \to 34 \to 31 \to 28 \to 26 \to 23 \to 21 \to 18 \to 17 \to 14 \\ \to 13 \to 10 \to 9 \to 6 \to 6 \to 3 \to 3 \to 0$$
.

In total, Vasya would eat 39 candies, while Petya — 29.

C. Sum of Cubes

2 seconds, 256 megabytes

You are given a positive integer x. Check whether the number x is representable as the sum of the cubes of two positive integers.

Formally, you need to check if there are two integers a and b ($1 \leq a, b$) such that $a^3+b^3=x$.

For example, if $x = 35$, then the numbers $a=2$ and $b=3$ are suitable ($2^3+3^3=8+27=35$). If $x=4$, then no pair of numbers a and b is suitable.

Input

The first line contains one integer t ($1 \leq t \leq 100$) — the number of test cases. Then t test cases follow.

Each test case contains one integer x ($1 \leq x \leq 10^{12}$).

Please note, that the input for some test cases won't fit into 32-bit integer type, so you should use at least 64-bit integer type in your programming language.

Output

For each test case, output on a separate line:

- "YES" if x is representable as the sum of the cubes of two positive integers.
- "NO" otherwise.

You can output "YES" and "NO" in any case (for example, the strings yEs, yes, Yes and YES will be recognized as positive).

input

```
7
1
2
4
34
35
16
703657519796
```

output

```
NO
YES
NO
NO
YES
YES
YES
```

The number 1 is not representable as the sum of two cubes.

The number 2 is represented as $1^3 + 1^3$.

The number 4 is not representable as the sum of two cubes.

The number 34 is not representable as the sum of two cubes.

The number 35 is represented as $2^3 + 3^3$.

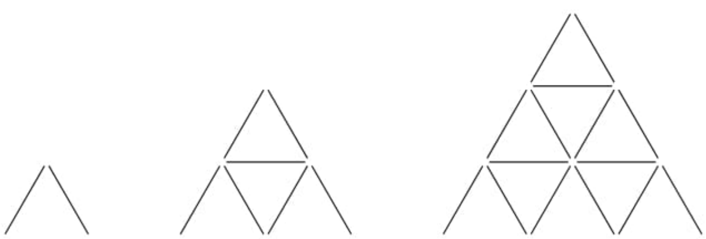
The number 16 is represented as $2^3 + 2^3$.

The number 703657519796 is represented as $5779^3 + 7993^3$.

D. Card Constructions

1 second, 256 megabytes

A card pyramid of height 1 is constructed by resting two cards against each other. For $h > 1$, a card pyramid of height h is constructed by placing a card pyramid of height $h - 1$ onto a base. A base consists of h pyramids of height 1, and $h - 1$ cards on top. For example, card pyramids of heights 1, 2, and 3 look as follows:



You start with n cards and build the tallest pyramid that you can. If there are some cards remaining, you build the tallest pyramid possible with the remaining cards. You repeat this process until it is impossible to build another pyramid. In the end, how many pyramids will you have constructed?

Input

Each test consists of multiple test cases. The first line contains a single integer t $(1 \leq t \leq 1000)$ — the number of test cases. Next t lines contain descriptions of test cases.

Each test case contains a single integer n $(1 \leq n \leq 10^9)$ — the number of cards.

It is guaranteed that the sum of n over all test cases does not exceed 10^9.

Output

For each test case output a single integer — the number of pyramids you will have constructed in the end.

input

```
5
3
14
15
24
1
```

output

```
1
2
1
3
0
```

In the first test, you construct a pyramid of height 1 with 2 cards. There is 1 card remaining, which is not enough to build a pyramid.

In the second test, you build two pyramids, each of height 2, with no cards remaining.

In the third test, you build one pyramid of height 3, with no cards remaining.

In the fourth test, you build one pyramid of height 3 with 9 cards remaining. Then you build a pyramid of height 2 with 2 cards remaining. Then you build a final pyramid of height 1 with no cards remaining.

In the fifth test, one card is not enough to build any pyramids.

E. Letters Shop

2 seconds, 256 megabytes

The letters shop showcase is a string s, consisting of n lowercase Latin letters. As the name tells, letters are sold in the shop.

Letters are sold one by one from the leftmost to the rightmost. Any customer can only buy some prefix of letters from the string s.

There are m friends, the i-th of them is named t_i. Each of them is planning to estimate the following value: how many letters (the length of the shortest prefix) would s/he need to buy if s/he wanted to construct her/his name of bought letters. The name can be constructed if each letter is presented in the equal or greater amount.

- For example, for s="arrayhead" and t_i="arya" 5 letters have to be bought ("**array**head").
- For example, for s="arrayhead" and t_i="harry" 6 letters have to be bought ("**arrayh**ead").
- For example, for s="arrayhead" and t_i="ray" 5 letters have to be bought ("**array**head").
- For example, for s="arrayhead" and t_i="r" 2 letters have to be bought ("**ar**rayhead").
- For example, for s="arrayhead" and t_i="areahydra" all 9 letters have to be bought ("**arrayhead**").

It is guaranteed that every friend can construct her/his name using the letters from the string s.

Note that the values for friends are independent, friends are only estimating them but not actually buying the letters.

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the length of showcase string s.

The second line contains string s, consisting of exactly n lowercase Latin letters.

The third line contains one integer m ($1 \le m \le 5 \cdot 10^4$) — the number of friends.

The i-th of the next m lines contains t_i ($1 \le |t_i| \le 2 \cdot 10^5$) — the name of the i-th friend.

It is guaranteed that $\sum\limits_{i=1}^{m} |t_i| \le 2 \cdot 10^5$.

Output

For each friend print the length of the shortest prefix of letters from s s/he would need to buy to be able to construct her/his name of them. The name can be constructed if each letter is presented in the equal or greater amount.

It is guaranteed that every friend can construct her/his name using the letters from the string s.

input

```
9
arrayhead
5
arya
harry
ray
r
areahydra
```

output

```
5
6
5
2
9
```

F. They Are Everywhere

2 seconds, 256 megabytes

Sergei B., the young coach of Pokemons, has found the big house which consists of n flats ordered in a row from left to right. It is possible to enter each flat from the street. It is possible to go out from each flat. Also, each flat is connected with the flat to the left and the flat to the right. Flat number 1 is only connected with the flat number 2 and the flat number n is only connected with the flat number $n - 1$.

There is exactly one Pokemon of some type in each of these flats. Sergei B. asked residents of the house to let him enter their flats in order to catch Pokemons. After consulting the residents of the house decided to let Sergei B. enter one flat from the street, visit several flats and then go out from some flat. But they won't let him visit the same flat more than once.

Sergei B. was very pleased, and now he wants to visit as few flats as possible in order to collect Pokemons of all types that appear in this house. Your task is to help him and determine this minimum number of flats he has to visit.

Input

The first line contains the integer n ($1 \le n \le 100\,000$) — the number of flats in the house.

The second line contains the row s with the length n, it consists of uppercase and lowercase letters of English alphabet, the i-th letter equals the type of Pokemon, which is in the flat number i.

Output

Print the minimum number of flats which Sergei B. should visit in order to catch Pokemons of all types which there are in the house.

input

```
3
AaA
```

output

```
2
```

input

```
7
bcAAcbc
```

output

```
3
```

input

```
6
aaBCCe
```

output

```
5
```

In the first test Sergei B. can begin, for example, from the flat number 1 and end in the flat number 2.

In the second test Sergei B. can begin, for example, from the flat number 4 and end in the flat number 6.

In the third test Sergei B. must begin from the flat number 2 and end in the flat number 6.