Computer Practicum 1: Practice C Exercises (Average to Difficult)
Fall Semester AY 2022-2023

Instructions: There are many problems in this document. Solve as many problems as you can. Use a separate file for each problem. The problems are labeled 00 to 07. When naming your files, use the format surname_firstname_problemXX.c (example: `Name_Surname_problem00.c`). When done, submit a pull request to my (Jordan's) CP1Preps repository with your submitted code.

The goal is to practice and enhance your skills using a variety of problems and approaches. You can always google (or Chat-GPT the answers) - but this doesn't give you enough practice to be an independent programmer. Take the time to solve and enhance your skills. Goodluck!

Note: Some of the problems were based from Codeforces.com

Proceed to the next page for the set of problems.

Problem 00.c

You are given the array a consisting of n positive (greater than zero) integers.

In one move, you can choose two indices i and j ($i \neq j$) such that the absolute difference between a_i and a_j is no more than one ($|a_i - a_j| \leq 1$) and remove the smallest of these two elements. If two elements are equal, you can remove any of them (but exactly one).

Your task is to find if it is possible to obtain the array consisting of **only one element** using several (possibly, zero) such moves or not.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \leq t \leq 1000$) — the number of test cases. Then t test cases follow.

The first line of the test case contains one integer n ($1 \leq n \leq 50$) — the length of a. The second line of the test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 100$), where a_i is the i-th element of a.

Output

For each test case, print the answer: "`YES`" if it is possible to obtain the array consisting of **only one element** using several (possibly, zero) moves described in the problem statement, or "`NO`" otherwise.

Example

input Copy

```
5
3
1 2 2
4
5 5 5 5
3
1 2 4
4
1 3 4 4
1
100
```

output Copy

```
YES
YES
NO
NO
YES
```

Problem 01.c

There is a house with n flats situated on the main street of Berlatov. Vova is watching this house every night. The house can be represented as an array of n integer numbers $a_1, a_2, \ldots, a_n$, where $a_i = 1$ if in the i-th flat the light is on and $a_i = 0$ otherwise.

Vova thinks that people in the i-th flats are disturbed and cannot sleep if and only if $1 < i < n$ and $a_{i-1} = a_{i+1} = 1$ and $a_i = 0$.

Vova is concerned by the following question: what is the minimum number k such that if people from exactly k pairwise distinct flats will turn off the lights then nobody will be disturbed? Your task is to find this number k.

Input

The first line of the input contains one integer n ($3 \le n \le 100$) — the number of flats in the house.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($a_i \in \{0, 1\}$), where a_i is the state of light in the i-th flat.

Output

Print only one integer — the minimum number k such that if people from exactly k pairwise distinct flats will turn off the light then nobody will be disturbed.

Examples

input

```
10
1 1 0 1 1 0 1 0 1 0
```

output

```
2
```

input

```
5
1 1 0 0 0
```

output

```
0
```

input

```
4
1 1 1 1
```

output

```
0
```

Problem 02.c

Imagine that you have a twin brother or sister. Having another person that looks exactly like you seems very unusual. It's hard to say if having something of an alter ego is good or bad. And if you do have a twin, then you very well know what it's like.

Now let's imagine a typical morning in your family. You haven't woken up yet, and Mom is already going to work. She has been so hasty that she has nearly forgotten to leave the two of her darling children some money to buy lunches in the school cafeteria. She fished in the purse and found some number of coins, or to be exact, n coins of arbitrary values $a_1, a_2, \ldots, a_n$. But as Mom was running out of time, she didn't split the coins for you two. So she scribbled a note asking you to split the money equally.

As you woke up, you found Mom's coins and read her note. "But why split the money equally?" — you thought. After all, your twin is sleeping and he won't know anything. So you decided to act like that: pick for yourself some subset of coins so that the sum of values of your coins is **strictly larger** than the sum of values of the remaining coins that your twin will have. However, you correctly thought that if you take too many coins, the twin will suspect the deception. So, you've decided to stick to the following strategy to avoid suspicions: you take the **minimum number of coins**, whose sum of values is strictly more than the sum of values of the remaining coins. On this basis, determine what **minimum** number of coins you need to take to divide them in the described manner.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of coins. The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 100$) — the coins' values. All numbers are separated with spaces.

Output

In the single line print the single number — the minimum needed number of coins.

Examples

input

```
2
3 3
```

output

```
2
```

input

```
3
2 1 2
```

output

```
2
```

Problem 03.c

Little time is left before Berland annual football championship. Therefore the coach of team "Losewille Rangers" decided to resume the practice, that were indefinitely interrupted for uncertain reasons. Overall there are n players in "Losewille Rangers". Each player on the team has a number — a unique integer from 1 to n. To prepare for the championship, the coach Mr. Floppe decided to spend some number of practices.

Mr. Floppe spent some long nights of his holiday planning how to conduct the practices. He came to a very complex practice system. Each practice consists of one game, all n players of the team take part in the game. The players are sorted into two teams in some way. In this case, the teams may have different numbers of players, but each team must have at least one player.

The coach wants to be sure that after the series of the practice sessions each pair of players had at least one practice, when they played in different teams. As the players' energy is limited, the coach wants to achieve the goal in the least number of practices.

Help him to schedule the practices.

Input

A single input line contains integer n ($2 \le n \le 1000$).

Output

In the first line print m — the minimum number of practices the coach will have to schedule. Then print the descriptions of the practices in m lines.

In the i-th of those lines print f_i — the number of players in the first team during the i-th practice ($1 \le f_i < n$), and f_i numbers from 1 to n — the numbers of players in the first team. The rest of the players will play in the second team during this practice. Separate numbers on a line with spaces. Print the numbers of the players in any order. If there are multiple optimal solutions, print any of them.

Examples

input

```
2
```

output

```
1
1 1
```

input

```
3
```

output

```
2
2 1 2
1 1
```

Problem 04.c

Prof. Vasechkin wants to represent positive integer n as a sum of addends, where each addends is an integer number containing only `1`s. For example, he can represent 121 as 121=111+11+–1. Help him to find the least number of digits `1` in such sum.

Input

The first line of the input contains integer n ($1 \le n < 10^{15}$).

Output

Print expected minimal number of digits `1`.

Examples

input

```
121
```

output

```
6
```

Problem 05.c

There are n points on a coordinate axis OX. The i-th point is located at the integer point x_i and has a speed v_i. It is guaranteed that no two points occupy the same coordinate. All n points move with the constant speed, the coordinate of the i-th point at the moment t (t **can be non-integer**) is calculated as $x_i + t \cdot v_i$.

Consider two points i and j. Let $d(i, j)$ be the minimum possible distance between these two points over any possible moments of time (even **non-integer**). It means that if two points i and j coincide at some moment, the value $d(i, j)$ will be 0.

Your task is to calculate the value $\sum_{1 \le i < j \le n} d(i, j)$ (the sum of minimum distances over all pairs of points).

Input

The first line of the input contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the number of points.

The second line of the input contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le 10^8$), where x_i is the initial coordinate of the i-th point. It is guaranteed that all x_i are distinct.

The third line of the input contains n integers $v_1, v_2, \ldots, v_n$ ($-10^8 \le v_i \le 10^8$), where v_i is the speed of the i-th point.

Output

Print one integer — the value $\sum_{1 \le i < j \le n} d(i, j)$ (the sum of minimum distances over all pairs of points).

Examples

input

```
3
1 3 2
-100 2 3
```

output

```
3
```

input

```
5
2 1 4 3 5
2 2 2 3 4
```

output

```
19
```

input

```
2
2 1
-3 0
```

output

```
0
```

Problem 06.c

Monocarp had a sequence a consisting of $n + m$ integers $a_1, a_2, \ldots, a_{n+m}$. He painted the elements into two colors, red and blue; n elements were painted red, all other m elements were painted blue.

After painting the elements, he has written two sequences $r_1, r_2, \ldots, r_n$ and $b_1, b_2, \ldots, b_m$. The sequence r consisted of all red elements of a **in the order they appeared in** a; similarly, the sequence b consisted of all blue elements of a **in the order they appeared in** a **as well**.

Unfortunately, the original sequence was lost, and Monocarp only has the sequences r and b. He wants to restore the original sequence. In case there are multiple ways to restore it, he wants to choose a way to restore that maximizes the value of

$$f(a) = \max(0, a_1, (a_1 + a_2), (a_1 + a_2 + a_3), \ldots, (a_1 + a_2 + a_3 + \cdots + a_{n+m}))$$

Help Monocarp to calculate the maximum possible value of $f(a)$.

Input

The first line contains one integer t ($1 \le t \le 1000$) — the number of test cases. Then the test cases follow. Each test case consists of four lines.

The first line of each test case contains one integer n ($1 \le n \le 100$).

The second line contains n integers $r_1, r_2, \ldots, r_n$ ($-100 \le r_i \le 100$).

The third line contains one integer m ($1 \le m \le 100$).

The fourth line contains m integers $b_1, b_2, \ldots, b_m$ ($-100 \le b_i \le 100$).

Output

For each test case, print one integer — the maximum possible value of $f(a)$.

Example

input

```
4
4
6 -5 7 -3
3
2 3 -4
2
1 1
4
10 -3 2 2
5
-1 -2 -3 -4 -5
5
-1 -2 -3 -4 -5
1
0
1
0
```

output

```
13
13
0
0
```

Problem 07.c

Young wilderness explorers set off to their first expedition led by senior explorer Russell. Explorers went into a forest, set up a camp and decided to split into groups to explore as much interesting locations as possible. Russell was trying to form groups, but ran into some difficulties...

Most of the young explorers are inexperienced, and sending them alone would be a mistake. Even Russell himself became senior explorer not long ago. Each of young explorers has a positive integer parameter e_i — his inexperience. Russell decided that an explorer with inexperience e can only join the group of e or more people.

Now Russell needs to figure out how many groups he can organize. It's not necessary to include every explorer in one of the groups: some can stay in the camp. Russell is worried about this expedition, so he asked you to help him.

Input

The first line contains the number of independent test cases T ($1 \le T \le 2 \cdot 10^5$). Next $2T$ lines contain description of test cases.

The first line of description of each test case contains the number of young explorers N ($1 \le N \le 2 \cdot 10^5$).

The second line contains N integers $e_1, e_2, \ldots, e_N$ ($1 \le e_i \le N$), where e_i is the inexperience of the i-th explorer.

It's guaranteed that sum of all N doesn't exceed $3 \cdot 10^5$.

Output

Print T numbers, each number on a separate line.

In i-th line print the maximum number of groups Russell can form in i-th test case.

Example

input

```
2
3
1 1 1
5
2 3 1 2 2
```

output

```
3
2
```