Superior Characters

Given a string, a character is said to be *superior* if it has two neighboring letters that are strictly smaller than itself. We compare characters by their location in the alphabet.

More formally, we say that the character at the i^{th} position is *superior* if a character exists at the $(i-1)^{\text{th}}$ position and $(i+1)^{\text{th}}$ position, and the character at the i^{th} position is strictly greater than the character at both $(i-1)^{\text{th}}$ and $(i+1)^{\text{th}}$ positions.

Given the frequencies of the 26 lowercase English letters, form a string using all these characters, such that the resultant string has the maximum number of superior characters. You need to print the maximum number of superior characters.

Complete the function `maximumSuperiorCharacters` which takes in an array of 26 integers denoting the frequencies of the English letters and returns an integer denoting the maximum number of superior characters.

Input Format

The first line contains an integer t, denoting the number of test cases.

Each of the next t lines contains 26 space-separated integers denoting the frequencies of the characters from `a` to `z`.

Constraints

- $1 \leq t \leq 10^4$
- $0 \leq \text{frequency} \leq 10^9$.

Output Format

Print the maximum number of superior characters that can be present in any string formed by using all the given characters.

Sample Input 0

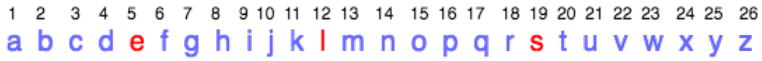
```
5
0 0 0 0 2 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1 2 2 3 4 0 3 4 4 1 3 1 4 4 1 0 0 0 0 0 4 2 3 2 2 1
1 1 3 3 1 1 4 4 3 1 3 3 3 3 0 1 2 0 4 2 1 3 0 3 1 1 1
3 3 0 2 2 2 4 1 2 1 1 1 3 3 0 0 3 2 2 4 1 4 4 1 2 1
2 1 4 1 0 2 0 3 1 2 0 3 1 1 2 0 1 4 2 3 2 3 2 0 2 1
```

Sample Output 0

```
1
25
24
25
21
```

Explanation 0

In the first test case, we have two **e**s, one **l**, one **s**, and none of the other letters. We can form a string with one superior character, for example, **else**. One can also show that this is the maximum number of superior characters that can be present. Thus, the answer is 1. There are also many other permutations of characters possible to form a string with maximum superior characters.



With the given frequency of 2 **e**s, 1 **l** and 1 **s**,

Permutations	Number of superior characters
e l s e	1
e e l s	0
e s l e	1
e e s l	1
e l e s	1
e s e l	1
l s e e	1
l e e s	0
l e s e	1
s e e l	0
s l e e	0
s e l e	1