D. Running Miles

time limit per test: 2 seconds
memory limit per test: 256 megabytes

There is a street with n sights, with sight number i being i miles from the beginning of the street. Sight number i has beauty b_i. You want to start your morning jog l miles and end it r miles from the beginning of the street. By the time you run, you will see sights you run by (including sights at l and r miles from the start). You are interested in the 3 most beautiful sights along your jog, but every mile you run, you get more and more tired.

So choose l and r, such that there are at least 3 sights you run by, and the sum of beauties of the 3 most beautiful sights minus the distance in miles you have to run is maximized. More formally, choose l and r, such that $b_{i_1} + b_{i_2} + b_{i_3} - (r - l)$ is maximum possible, where i_1, i_2, i_3 are the indices of the three maximum elements in range $[l, r]$.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each test case contains a single integer n ($3 \le n \le 10^5$).

The second line of each test case contains n integers b_i ($1 \le b_i \le 10^8$) — beauties of sights i miles from the beginning of the street.

It's guaranteed that the sum of all n does not exceed 10^5.

Output

For each test case output a single integer equal to the maximum value $b_{i_1} + b_{i_2} + b_{i_3} - (r - l)$ for some running range $[l, r]$.

Example

input Copy

```
4
5
5 1 4 2 3
4
1 1 1 1
6
9 8 7 6 5 4
7
100000000 1 100000000 1 100000000 1 100000000
```

output Copy

```
8
1
22
299999996
```

Note

In the first example, we can choose l and r to be 1 and 5. So we visit all the sights and the three sights with the maximum beauty are the sights with indices 1, 3, and 5 with beauties 5, 4, and 3, respectively. So the total value is $5 + 4 + 3 - (5 - 1) = 8$.

In the second example, the range $[l, r]$ can be $[1, 3]$ or $[2, 4]$, the total value is $1 + 1 + 1 - (3 - 1) = 1$.