

JUMPING JAPANG

There are **N** tiles numbered from **1** to **N** and each tile has a integer number (negative or positive in the range -1000000 to 1000000) written on it. You are standing at tile 0 where there is no number written. You have to reach tile **N** using only three kinds of moves.

If you are standing on tile **x** you can move to:

->Tile **x+1** or

->Tile **x+2** or

->Tile **x+3**

you cannot visit a tile more than once. And you cannot go beyond tile **N**.

Initially you have a **sum** set to **0**. Now when ever you visit a tile you add the number written on that tile to **sum**. Now you have to reach the **N**th tile with maximum **sum**.

You dont have to tell the path but you have to report what is the maximum **sum** you can achieve along the way.

INPUT:

First line contains **T** ≤ 100 , the number of test cases.

Every test case has two lines of input which is as follows:

The first line contains an integer **N** ($N \leq 1000$) the size of the array.

The second line contains the **N** values denoting the number on each tile in order from 1 to **N**.

OUTPUT:

For every test case output the maximum sum in a new line

EXAMPLE INPUT:

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

EXAMPLE OUTPUT:

```
15
1
```