Final Destination -ABII

Problem statement :

Assume that you are in front of Academic Block Ground Floor-005.(**ABII-005**)
There are four types of queries:

```
Type-1 : move one floor up.

Type-2 : move one floor down.

Type-3 : move one class left.

Type-4 : move one class right.
```

You are given a list of queries after following the queries you have to return the class number in front of you.
Note:
1.If you are at ground floor and the query is to move one floor down then you remain at ground floor only.
2.If you are at 3rd floor and the query is to move one floor up then you remain at 3rd floor.
3.If you are at 101 and query is to move left you will move to 114.
4.If you are at 114 and query is to move right you will move to 101.
5.Print **ABII-10** do NOT **ABII-010.**

Example1:

Queries : 1 3 2 4 4 4 4 4 4
Output **ABII-010**
Explanation:

```
intitially at 005

1 one floor up   -> 105

3 move left      -> 104

2 one floor down -> 004

4 move right     -> 005

4 move right     -> 006

4 move right     -> 007

4 move right     -> 008
```

```
4 move right    -> 009

4 move right    -> 010

OUTPUT ABII-10
```

Input Format

- First line of input contains no of testcases **T**
- Next 2*T lines contains T testcases
- First line of each testcase contain size of queries **N**
- Second line of each testcase contain N elements array

Constraints

- $1 <= T <= 100$
- $1 <= N <= 10^6$
- $1 <= Q[i] <= 4$
- *Note:*
- Assume if you are at class 113 if a query come's Type-1 you will jump to class 213

Output Format

- For each testcase output the destination class you reach

Sample Input 0

```
1

5

1 1 1 3 3
```

Sample Output 0

```
ABII-303
```

Explanation 0

Testcase0:
Queries : 1 1 1 3 3
Output **ABII-303**
Explanation:

```
Intitially   at 005

1 one floor up -> 105

1 one floor up -> 205

1 one floor up -> 305

3 move left    -> 304

3 move left    -> 303

OUTPUT ABII-303
```

Sample Input 1

```
2

9

1 3 2 4 4 4 4 4 4

3

1 1 3
```

Sample Output 1

```
ABII-10

ABII-204
```

Solution in C:

```
#include <string.h>
#include <math.h>
#include <stdlib.h>
int call(int q[],int n){
   int cls=5,i,r,c;
      for(i=0;i<n;i++){
         switch (q[i]){
            case 1:
               if(cls+100<400){
                  cls+=100;
               }
               break;
            case 2:
               if(cls-100>0){
                  cls-=100;
               }
               break;
            case 3:
               r=cls%100;
               if(r>2){
                  cls-=1;
               }else {
                  cls +=13;
               }
               break;
            case 4:
               c=cls%100;
               if(c<14){
                  cls+=1;
               }else{
```

```
                cls-=13;
            }
            break;

        }
    }
    return cls;

}
int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int t;
    scanf("%d",&t);
    while(t--){
        int n,i;
        scanf("%d",&n);
        int a[n];
        for(i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        printf("ABII-%d\n",call(a,n));
    }
    return 0;
}
```

Solution in Python:

```
test=int(input())
for tests in range(test):
    q=int(input())
    a=list(map(int,input().split()))
    pos=5
    for i in a:
        if((i==1 and pos>300) or (i==2 and pos<100)):
            pos=pos
        elif(i==1):
            pos+=100
        elif(i==2):
            pos-=100
        elif(i==3 and pos%100==1):
            pos+=13
        elif(i==4 and pos%100==14):
            pos-=13
        elif(i==3):
            pos-=1
        elif(i==4):
            pos+=1
    print("ABII-"+str(pos))
```

Solution in Java:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scan= new Scanner(System.in);
        int t=scan.nextInt();
        while(t-- >0){
            int n=scan.nextInt();
            int[] q= new int[n];
            for(int i=0;i<n;i++){
                q[i]=scan.nextInt();
            }
            int res=0;
            for(int i=0;i<n;i++){
                res=solve(q,n);

            }
            System.out.println("ABII-"+res);

        }
    }
    public static int solve(int[] q ,int n){
        int cls=5;
        for(int i=0;i<n;i++){
            switch (q[i]){
```

```
            case 1->{
                if(cls+100<400){
                    cls+=100;
                }
            }
            case 2->{
               if(cls-100>0){
                   cls-=100;
               }
            }
            case 3->{
               int r=cls%100;
               if(r>2){
                   cls-=1;
               }else {
                   cls +=13;
               }
            }
            case 4->{
               int c=cls%100;
               if(c<14){
                   cls+=1;
               }else{
                   cls-=13;
               }
            }

        }
    }
    return cls;
```

```
    }
  }
```