**Next in Series**

[maths](http://www.practice.geeksforgeeks.org/tag-page.php?tag=maths&isCmp=0)[progression](http://www.practice.geeksforgeeks.org/tag-page.php?tag=progression&isCmp=0)[series](http://www.practice.geeksforgeeks.org/tag-page.php?tag=series&isCmp=0)[Inmobi](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Inmobi&isCmp=1)

Given an array of integers, the task is to check whether it's an arithmetic progression, geometric progression or fibonacci series, and need to print the next number in the series or -99999 if the given numbers does not follow any of the above series. The fibonacci series need not start with 0.   A valid fibonacci series could be 8,9,17

**NOTE**: In case of clash, following precedence shold be followed: AP>GP>FIBO

**Input:**  
The First line of the input contains a single interger T denoting the number of testcases. Then T test cases follow. Each test cases consists of two lines. First tline of each test case contains an interger N denoting the no. of integers in an array. The second line of each test case contains N integers denoting the array elements.  
  
**Output:**  
Corresponding to each test case, print the desired output modulo 10^9+7 in a new line.

NOTE: In case of floating point answer , the answer should be next greater integer.  
  
**Constraints:**  
1<=T<=100  
2<=N<=20  
0<=A[i]<=10^9

**Example:**  
**INPUT:**  
4  
3  
1 2 3  
4  
1 3 9 27  
4  
2 3 5 8  
3  
2 5 6

**OUTPUT:**  
4  
81  
13  
-99999

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=781>

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

int main(){

    int t;

    scanf("%d", &t);

    while(t-- ) {

       int n;

       scanf("%d", &n);

       int arr[n];

       for(int i =0; i<n; i++) {

         scanf("%d", &arr[i]);

       }

       std::vector<int> dif;

       std::vector<double> geom;

       for(int i =0; i + 1 < n; i++) {

          dif.push\_back(arr[i+1] - arr[i]);

          geom.push\_back((double) arr[i+1] / (double) arr[i]);

       }

      // bool sec = false;

       if(dif[1] == dif[0]) {

          printf("%d**\n**", arr[n-1] + dif[0]);

        //  sec=true;

       } else if (geom[1] == geom[0]) {

          printf("%d**\n**", arr[n-1] \* (int) geom[0]);

         // sec = true;

//tienen que ser distintos porque por ejemplo si todos son ceros entra a este if

       }else if (n - 3 >= 0 && arr[n-3] != arr[n-2] && arr[n-2] != arr[n-1] &&

                arr[n - 3] + arr[n - 2] == arr[n - 1]) {

          printf("%d**\n**", arr[n-1] + arr[n-2]);

        //  sec = true;

       }else{

          printf("%d**\n**", -99999);

       }

    }

 return 0;

}