

Ягоди

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Цвети е на *СДА*. Асистентите са донесли n купички с ягоди, като в купичка с номер i има a_i ягоди. Всички ягоди са номерирани с последователни числа: в първата купичка ягодите били номерирани от 1 до a_1 , във втората - от $a_1 + 1$ до $a_1 + a_2$ и така нататък.

Асистентите са измислили състезание: те казват номер на някоя вкусна ягода, а който първи познае в коя купичка се намира ягодата, я получава.

Цвети обожава ягоди и иска да изяде всички :). Помогнете й!

Input Format

Първият ред от стандартния изход съдържа числото n - броя на купичките.

На следващия ред следват n числа - a_1, a_2, \dots, a_n , където a_i е броят ягоди в i -тата купичка.

Третият ред съдържа числото m - броя на вкусните ягоди, посочени от асистентите.

На четвъртия ред имаме m числа - q_1, q_2, \dots, q_n - номерата на вкусните ягоди.

Constraints

$$1 \leq n \leq 100000$$

$$1 \leq m \leq 100000$$

$$1 \leq a_i \leq 10^9$$

$$1 \leq q_i \leq a_1 + a_2 + \dots + a_n$$

Output Format

Изведете m реда на стандартния изход. Ред номер i съдържа номера на купичката в която се намира ягода номер q_i .

Sample Input 0

```
5
2 7 3 4 9
3
1 25 11
```

Sample Output 0

```
1
5
3
```

Explanation 0

Първата купичка съдържа ягодите с номера 1 и 2, втората - ягодите с номера 3,4,5,6,7,8,9 и така нататък. Така получаваме, че първата ягода е в първата купичка, 25-тата в петата, а 11-тата - в третата.

Current Buffer (saved locally, editable)

C++

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
int binarySearch(long int* arr,long int left,long int right,long int element){
    long int mid=0;
    if(right>=left){
        int mid=(left+right)/2;
        if(arr[mid]==element){
            return mid+1;
        }
        if(arr[mid]>element){
            return binarySearch(arr,left,mid-1,element);
        }
        if(arr[mid]<element){
            return binarySearch(arr,mid+1,right,element);
        }
    }

    return left+1;
}

int main() {
    int cupsNumber;
    cin>>cupsNumber;
    int currentCup;
    long int lastCup=0;
    long int* strawberries=new long int[cupsNumber];

    for(int i=0;i<cupsNumber;i++){
        cin>>currentCup;
        strawberries[i]=currentCup+lastCup;
        lastCup=strawberries[i];
    }

    int delStrawberriesNumber;
    cin>>delStrawberriesNumber;
    long int* delStrawberries=new long int[delStrawberriesNumber];
    for(int i=0;i<delStrawberriesNumber;i++){
        cin>>delStrawberries[i];
    }

    int* answers=new int[delStrawberriesNumber];
    for(int i=0;i<delStrawberriesNumber;i++){
        cout<<binarySearch(strawberries,0,cupsNumber,delStrawberries[i])<<endl;
    }

    return 0;
}
```

Line: 24 Col: 1