Problem: *The Birthday Bar*

Lily has a chocolate bar consisting of a row of squares where each square has an integer written on it. She wants to share it with Ron for his birthday, which falls on month and day . Lily wants to give Ron a piece of chocolate only if it contains consecutive squares whose integers sum to .

Given , , and the sequence of integers written on each square of Lily's chocolate bar, how many different ways can Lily break off a piece of chocolate to give to Ron?

For example, if , and the chocolate bar contains rows of squares with the integers written on them from left to right, the following diagram shows two ways to break off a piece:


```
1 + 2 = 3
```

```
1   2   1   3   2
```

```
2 + 1 = 3
```
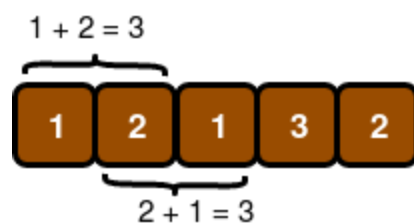
Input Format

The first line contains an integer denoting (the number of squares in the chocolate bar).
The second line contains space-separated integers describing the respective values of (the numbers written on each consecutive square of chocolate).
The third line contains two space-separated integers describing the respective values of (Ron's birth *day*) and (Ron's birth *month*).

Constraints

-
- , where ()
-
-

Output Format

Print an integer denoting the total number of ways that Lily can give a piece of chocolate to Ron.

Sample Input 0

```
5
1 2 1 3 2
3 2
```

Sample Output 0

2

Explanation 0

This sample is already explained in the problem statement.

Sample Input 1

```
6
1 1 1 1 1 1
3 2
```

Sample Output 1

```
0
```

Explanation 1

Lily only wants to give Ron consecutive squares of chocolate whose integers sum to . There are no possible pieces satisfying these constraints:

Thus, we print as our answer.

Sample Input 2

```
1
4
4 1
```

Sample Output 2

```
1
```

Explanation 2

Lily only wants to give Ron square of chocolate with an integer value of . Because the only square of chocolate in the bar satisfies this constraint, we print as our answer.

Solution:

```cpp
int main() {
    int squares;
    int date, month;
    int possibleWays=0;

    cin>>squares;
    int values[squares];

    /*Feeding the data*/
    for(int i=0; i<squares; i++)
      { cin>>values[i]; }

    cin>>date >>month;

    /*Counting the possible ways*/
    for(int i=0; i<squares; i++)
      {
         int sum=0;
         for(int j=0; j<month && i+(month-1)<squares ; j++)
           { sum+=values[i+j]; }
         if(sum==date)
           {possibleWays+=1;
           }
      }

    cout<<possibleWays;

    return 0;
}
```