Problem Statement

An array of integers is given where each member represents the maximum number of steps that can be made forward from that member. Write a program to print the minimum number of hops required to reach the end of the array, starting from first element. Assume all the members are greater than or equal to 0. If a member is 0, then no forward movement is allowed through that.

Input and Output Format:

Read the input from standard input and print the output to standard output.

First line in the input is the number of elements of the array where second line has the array elements separated by space. Print "infinite" if last element is unreachable , minimum number of jumps otherwise.

Input and Output Format:

Read the input from standard input and print the output to standard output.

First line in the input is the number of elements of the array where second line has the array elements separated by space. Print "infinite" if last element is unreachable , minimum number of jumps otherwise.

Sample Input	Sample Output	Explanation
10 1398221010	3	Minimum hops : 1–> 3–>9–> 0
5 10101	infinite	No movement allowed after the second element

Languages: Java

StringTester.java | Main.java | Main1.java | Solution.java | DivisorSum.java

```java

public class MyCode {

    public static void main(String[] args ) {
        // TODO Auto-generated method stub
     Scanner sc=new Scanner(System.in);
     int size=sc.nextInt();
     int arr[]= new int[size];
     int jump=0;

     for(int i=0; i < size; i++) {
     arr[i]=sc.nextInt();
     }
      if(arr[1]==0 || arr[0]==0) {
          System.out.println("infinite");
      }
      else {

          for(int i=0; i<size; i++) {
              if(arr[i]!=arr.length-2 || arr[i]==0)
                  jump++;
              else
                  break;
          }
          System.out.println(jump);
      }
      sc.close();
    }
}


```

```
 9        Scanner sc=new Scanner(System.in);
10        int size=sc.nextInt();
11        int arr[]= new int[size];
12        int jump=0;
13
14        for(int i=0; i < size; i++) {
15        arr[i]=sc.nextInt();
16        }
17         if(arr[1]==0 || arr[0]==0) {
18             System.out.println("infinite");
19         }
20         else {
```

Console ×

<terminated> MyCode [Java Application] C:\Users\swastikv.k\Downloads\spring-tool

```
10
1 3 9 5 2 2 1 0 1 0
3
```

```
11          int arr[]= new int[size];
12          int jump=0;
13
14          for(int i=0; i < size; i++) {
15          arr[i]=sc.nextInt();
16          }
17           if(arr[1]==0 || arr[0]==0) {
18                System.out.println("infinite");
19           }
20           else {
```

Console x

<terminated> MyCode [Java Application] C:\Users\swastikv.k\Downloads\spri

```
5
1 0 1 0 1
infinite
```