Problem I. The sum of consecutive

You are given an array of integers a [1], a [2], ..., a [n] and natural numbers k and m.
Specify the minimum value i for which a [i] + a [i + 1] + ... + a [i + k] = m (that is, the sum of k + 1 consecutive array elements is equal to m).
If there is no such value, then output 0. Do not use nested loops.

Input data
The values n, k and m (m <= 10000, 0 <k <n <= 30000; n is the number of elements in the array) are first fed to the program.
The next line of the input data contains the elements of the array themselves - integers, modulo not exceeding 100.

Output
Print the answer to the problem.

Examples of
input data
4 1 22
4 numbers are entered, it is necessary to find the number of the beginning of the sum of (1 + 1) consecutive numbers and equal (sum) to the 22.

9 13 10 -11

output
1 (because 1 is order number of 9 in sequence 9 13 10 -11)

7 3 20
12 42 5 5 8 2 53

answer 2

w=0
12 42 5 5 8 2 53

w=1
12 42 5 5 8 2 53

w=2
12 42 5 5 8 2 53

The WORST

```
#include <iostream>
#include <vector>
using namespace std;

void sumofConsecutives(int numbers, int k, int sum) {
   int count = 0;
   int newNumber = 0;
   int arr[numbers];
   for (int i = 0; i < numbers; i++) {
       cin >> newNumber;
       arr[i] = newNumber;
   }
   int w = 0;
   int localSum;
   while (w < numbers - (k + 1)) {
       localSum = 0;
       for (int j = w + 0; j < w + k + 1; j++) {
           localSum += arr[j];
       }
       if (localSum == sum) {
           cout << w + 0 << endl;
           return;
       }
       w++;
   }
}
int main ()
{
   int numbers = 7;
   int k = 3;
   int sum = 20;
   sumofConsecutives(numbers, k, sum);
}
```

The BEST

```
void sumofConsecutives(int numbers, int k, int sum) {
   int newNumber = 0;
   int w = 0;
   int localSum = 0;
   int arr[k + 1];
   int a = 0;
   for (int i = 0; i < numbers; i++) {
       cin >> newNumber;
       if (i < k + 1) {
           localSum = localSum + newNumber;
           arr[a] = newNumber;
           a++;
       }
       else {
           localSum = localSum + newNumber - arr[0];
           for (int p = 0; p < k + 1; p++) {
               arr[p] = arr[p + 1];
           }
           arr[k] = newNumber;
       }
       if (localSum == sum) {
           cout << "answer: " << i - (k) << endl;
           return;
       }
   }

}
```

MIDDLE (As in task)

```
void sumofConsecutives(int numbers, int k, int sum) {
  int count = 0;
  int newNumber = 0;
  int arr[numbers];
  for (int i = 0; i < numbers; i++) {
      cin >> newNumber;
      arr[i] = newNumber;
  }
  int w = 0;
  int localSum = 0;
  while (w < numbers) {
      if (w < k + 1) {
          localSum += arr[w];
      }
      else {
          localSum = localSum + arr[w] - arr[w - (k +
1)];
      }
      if (localSum == sum) {
          cout << "answer: " << w - (k)<< endl;
          return;
      }
      w++;
  }
}
```