2.7

AC

```
//
// Daniel López Marqués
// A41
// A47
//

#include <iostream>
#include <fstream>
#include <vector>



// Definicion de la función lógica contarPositivos;

//contarPositivos=# i,j,k : 0<=i<j<v.size() ^ i<=k<=j : v[k]>0

// Definicion del predicado hayMasPositivos

//hayMasPositivos=exists m : x<m<=y ^ m==(y-x)/2 :
contarPositivos(x,m-1)>=contarPositivos(m,y)

// Especificación del problema

//Palg = {v.size()>0 ^ k%2==0}
//int contarPositivos(vector<int> v,int k) dev {int contador}
//Qalg= {contador==#p,q:0<=p<q<v.size() ^ q-p==k :hayMasPositivos(v,p,q)}

// Invariantes de cada uno de los bucles utilizados en la impleemntación del problema
//Pbucle1={v.size()>0 ^ k%2==0 ^positivosIzq==0 ^positivosDcha==0 ^ contador==0}
//Qbucle1={positivosIzq==#n,p,q:0<=n<(k/2):v[n]>0 ^positivosDcha==#n:(k/2)<=n<k:v[n]>0 ^
contador==#p,q:0<=p<q<k ^q-p==k :hayMasPositivos(v,p,q)}
//Inv1={positivosIzq==#n:0<=n<(k/2):v[n]>0 ^positivosDcha==#n:(k/2)<=n<k:v[n]>0 ^
contador==#p,q:0<=p<q<k ^q-p==k :hayMasPositivos(v,p,q)}

//Pbucle2={positivosIzq==#n:0<=n<(k/2):v[n]>0 ^positivosDcha==#n:(k/2)<=n<k:v[n]>0 ^
contador==#p,q:0<=p<q<k ^q-p==k :hayMasPositivos(v,p,q)}
//Qbucle2={positivosIzq==#n:0<=n<(k/2):v[n]>0 ^positivosDcha==#n:(k/2)<=n<k:v[n]>0 ^
contador==#p,q:0<=p<q<v.size() ^q-p==k :hayMasPositivos(v,p,q)}
//Inv2={positivosIzq==#n:0<=0<(k/2):v[n]>0 ^positivosDcha==#n:(k/2)<=n<k:v[n]>0 ^
contador==#p,q:0<=p<q<i ^q-p==k :hayMasPositivos(v,p,q)}

int contarPositivos(std::vector<int> const& v, int k) {
    int positivosIzq = 0, positivosDcha = 0, contador = 0;
    for (int i = 0; i < k; i++) {
        if (i < (k / 2) && v[i] > 0)positivosIzq++;
        else if (i >= (k / 2) && v[i] > 0)positivosDcha++;
    }
    if (positivosIzq >= positivosDcha) contador++;
    for (int i = k; i < v.size(); i++) {
        if (v[i - k] > 0) positivosIzq--;
        if (v[i - (k / 2)] > 0) {
            positivosDcha--;
            positivosIzq++;
```

0.25

0.25

0.25

0.2

1.75

No se calcula en el bucle 1.

mal apuntado.

```
        }
        if (v[i] > 0) positivosDcha++;
        if (positivosIzq >= positivosDcha) contador++;
    }
    return contador;
}

// resuelve un caso de prueba, leyendo de la entrada la
// configuración, y escribiendo la respuesta
bool resuelveCaso() {
    int numElems, K; // k tiene que ser par
    std::cin >> numElems;
    if (numElems == 0) return false;
    std::cin >> K;
    std::vector<int> v(numElems);
    for (int& i : v) std::cin >> i;

    std::cout<< contarPositivos(v,K)<<"\n";
    // LLamada a la función que resuelve el problema
    // Escribir el resultado



    return true;
}


int main() {

#ifndef DOMJUDGE
    std::ifstream in("datos1.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    while (resuelveCaso())
        ;

#ifndef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif

    return 0;
}
```