

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó y el Compa Mamado
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: //Mediana movil
11:
12: #include <iostream>
13: using namespace std;
14:
15: int main() {
16:     const float ceroKelvin = -273,15;
17:     const int TAMANIO = 1e6;
18:     double temperaturas[TAMANIO];
19:     double ordenados[TAMANIO];
20:     int posicion = 0, N = 3;
21:     double mediana;
22:
23:     cout << "Introduzca las temperaturas: \n";
24:     cin >> temperaturas[posicion];
25:
26:
27:     //Lectura de Datos
28:
29:     while(temperaturas[posicion] >= ceroKelvin){
30:         posicion++;
31:         cin >> temperaturas[posicion];
32:     }
33:     //Computo
34:
35:     double a_insertar;
36:     int k;
37:
38:     /*//Cálculo de la mediana
39:     Primero ordenamos las temperaturas y posteriormente hacemos su mediana*/
40:
41:     for(int iter = 0; iter < posicion - (N-1); iter++){
42:         //asignamos los N temperaturas al vector ordenador
43:         for(int i=iter; i < N+iter; i++)
44:             ordenados[i] = temperaturas[i];
45:         for (int j = iter; j < N+iter; j++){
46:             a_insertar = ordenados[j];
47:
48:             for (k = j; k > iter && a_insertar < ordenados[k-1]; k--) // Ordenación de menor a mayor
49:                 ordenados[k] = ordenados[k-1];
50:
51:             ordenados[k] = a_insertar;
52:         }
53:         /*//Comprobar ordenacion
54:         cout << "El vector ordenado es: ";
55:         for (int j = iter; j < N+iter; j++)
56:             cout << ordenados[j] << " ";
57:         */
58:         mediana = 0;
59:         if(N % 2 != 0)
60:             mediana = ordenados[iter+(N/2)];
61:         else
62:             mediana = (ordenados[iter+(N/2)] + ordenados[iter+(N/2)-1]) / 2;
63:
64:         cout << " " << mediana;
65:     }
66:
67:     cout << " " << temperaturas[posicion];
68: }
69:

```

```
1: /**
2:  * @file mediana_movil.cpp
3:  * @brief Calcula la mediana movil de una secuencia de temperaturas
4:  *
5:  * @author Fulanito...
6:  * @date Diciembre-2020
7:  *
8:  * Escriba un programa que procesa una secuencia de valores de temperatura hasta que
9:  * se introduce una temperatura menor que el cero absoluto (-273.15 grados).
10: *
11: * Como resultado, escribirá una secuencia de datos que corresponde a la mediana móvil
12: * con tamaño N. Cada valor de la secuencia de esta mediana móvil corresponde a:
13: *
14: *   - La mediana de los primeros N datos desde el 0 al N-1,
15: *   - la mediana de los N siguientes desde el 1 al N,
16: *   - la mediana de los N siguientes desde el 2 al N+1,
17: *   - etc.
18: *
19: * Por tanto, si hay D datos, la secuencia resultado tendrá D-(N-1) valores. Esta
20: * secuencia, además, estará también terminada con un valor centinela.
21: *
22: * El problema se puede resolver cargando toda la secuencia de datos y luego
23: * calculando la mediana móvil para cada N datos, aunque también se podría limitar el
24: * tamaño de la memoria ocupada evitando tener toda la secuencia, pues sólo es
25: * necesario almacenar los últimos N valores.
26: *
27: * Nota: En el problema, puede suponer que N es fijo y tiene un valor
28: * predeterminado. Así, evita tener que introducirlo; tanto la entrada como la
29: * salida serán una simple secuencia.
30: *
31: * Un ejemplo de ejecución, con N valiendo 5, es:
32: *   1 2 3 4 5 6 7 8 9 -300
33: *   3 4 5 6 7 -300
34: * donde la primera línea es la entrada y la segunda la salida.
35: *
36: * Otra ejemplo, ahora con N valiendo 5, es:
37: *   -0.4 -0.5 -0.9 -0.6 -0.7 -0.1 0.9 0.7 0.2 1.2 1.5 -300
38: *   -0.6 -0.6 -0.6 -0.1 0.2 0.7 0.9 -300
39: *
40: * El mismo ejemplo, ahora con N valiendo 3, es:
41: *   -0.4 -0.5 -0.9 -0.6 -0.7 -0.1 0.9 0.7 0.2 1.2 1.5 -300
42: *   -0.5 -0.6 -0.7 -0.6 -0.1 0.7 0.7 0.7 1.2 -300
43: */
```