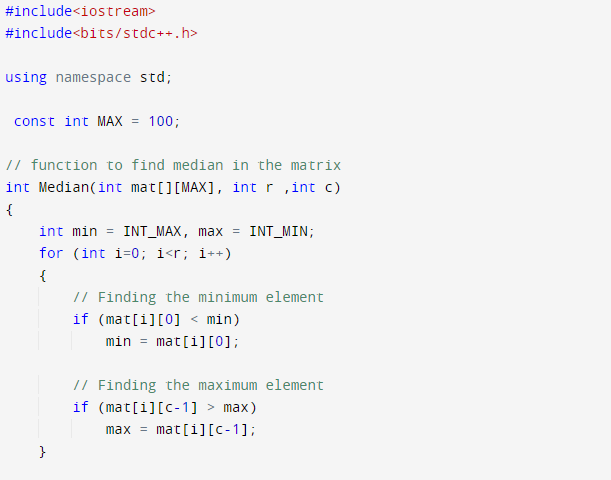
**DAA Assignment -1**

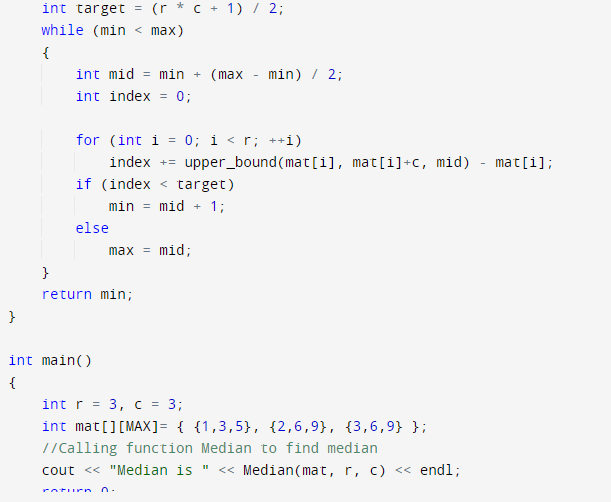
**(Implemented the following problems using C++)**

1 .Given a row wise sorted matrix of size **R\*C** where R and C are always **odd**, find the median of the matrix. **5Mark**

* **Constraints:**  
  1 <= R, C <= 400  
  1 <= matrix[i][j] <= 2000

***DRIVER CODE OF THE PROBLEM STATEMENT*:**





**Test Case 1:**

**Input**:

R = 3, C = 3

M = [[1, 3, 5],

  [2, 6, 9],

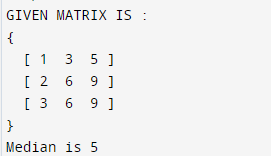
  [3, 6, 9]]

**Output:** 5

**Explanation**: Sorting matrix elements gives

us {1,2,3,3,5,6,6,9,9}. Hence, 5 is median.

**STD - OutPut:**

****

**Test Case 2:**

**Input:**

R = 3, C = 1

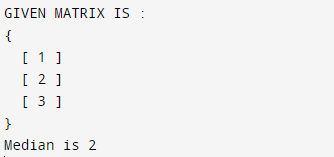
M = [[1], [2], [3]]

**Output:** 2

**Explanation**: Sorting matrix elements gives

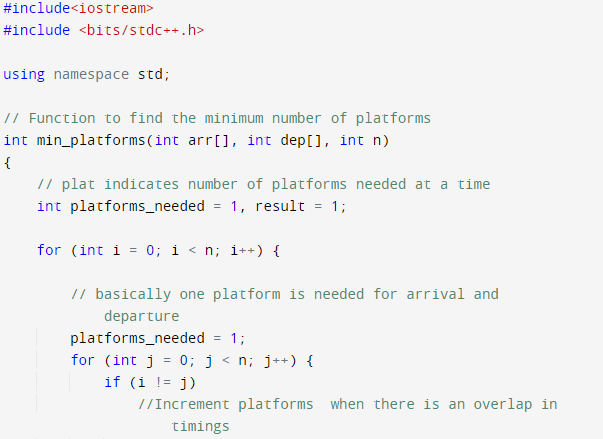
us {1,2,3}. Hence, 2 is median.

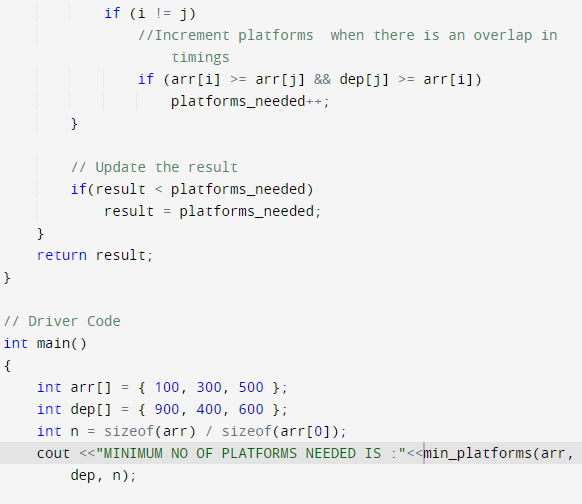
**STD - OutPut :**

****

2. Given the arrival and departure times of all trains that reach a railway station, the task is to find the minimum number of platforms required for the railway station so that no train waits. We are given two arrays that represent the arrival and departure times of trains that stop. **5Marks**

***DRIVER CODE FOR THE PROBLEM STATEMENT*:**

****

****

**Test case 1 :**

***Input:****arr[] = {9:00, 9:40, 9:50, 11:00, 15:00, 18:00}, dep[] = {9:10, 12:00, 11:20, 11:30, 19:00, 20:00}****Output:****3****Explanation:****There are at-most three trains at a time (time between 9:40 to 12:00)*

***STD OUTPUT:***

**

**Test case 2 :**

***Input:****arr[] = {9:00, 9:40}, dep[] = {9:10, 12:00}****Output:****1****Explanation:****Only one platform is needed.*

***STD OUTPUT*:**

****