Run | Debug | Stop | Share | Save | {} Beautify | Language C++

main.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;
 const int MAX = 100;
 int binaryMedian(int m[][MAX], int r ,int c)
{
    int min = INT_MAX, max = INT_MIN;
    for (int i=0; i<r; i++)
    {
        if (m[i][0] < min)
            min = m[i][0];
        if (m[i][c-1] > max)
            max = m[i][c-1];
    }
    int desired = (r * c + 1) / 2;
    while (min < max)
    {
        int mid = min + (max - min) / 2;
        int place = 0;
        for (int i = 0; i < r; ++i)
            place += upper_bound(m[i], m[i]+c, mid) - m[i];
        if (place < desired)
            min = mid + 1;
        else
            max = mid;
    }
    return min;
}
```

Run | Debug | Stop | Share | Save | { } Beautify | Language C++

main.cpp

```
    {
        if (m[i][0] < min)
            min = m[i][0];
        if (m[i][c-1] > max)
            max = m[i][c-1];
    }
    int desired = (r * c + 1) / 2;
    while (min < max)
    {
        int mid = min + (max - min) / 2;
        int place = 0;
        for (int i = 0; i < r; ++i)
            place += upper_bound(m[i], m[i]+c, mid) - m[i];
        if (place < desired)
            min = mid + 1;
        else
            max = mid;
    }
    return min;
}
int main()
{
    int r = 3, c = 3;
    int m[][MAX]= { {1,3,5}, {2,6,9}, {3,6,9} };
    cout << "Median is " << binaryMedian(m, r, c) << endl;
    return 0;
}
```

input

```
Median is 5


...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | { } Beautify | Language C++

main.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;
 const int MAX = 100;
 int binaryMedian(int m[][MAX], int r ,int c)
{
    int min = INT_MAX, max = INT_MIN;
    for (int i=0; i<r; i++)
    {
        if (m[i][0] < min)
            min = m[i][0];
        if (m[i][c-1] > max)
            max = m[i][c-1];
    }
    int desired = (r * c + 1) / 2;
    while (min < max)
    {
        int mid = min + (max - min) / 2;
        int place = 0;
        for (int i = 0; i < r; ++i)
            place += upper_bound(m[i], m[i]+c, mid) - m[i];
        if (place < desired)
            min = mid + 1;
        else
            max = mid;
    }
    return min;
}
```

Run | Debug | Stop | Share | Save | {} Beautify | Language C++

main.cpp

```
    {
        if (m[i][0] < min)
            min = m[i][0];
        if (m[i][c-1] > max)
            max = m[i][c-1];
    }
    int desired = (r * c + 1) / 2;
    while (min < max)
    {
        int mid = min + (max - min) / 2;
        int place = 0;
        for (int i = 0; i < r; ++i)
            place += upper_bound(m[i], m[i]+c, mid) - m[i];
        if (place < desired)
            min = mid + 1;
        else
            max = mid;
    }
    return min;
}
int main()
{
    int r = 3, c = 1;
    int m[][MAX]= { {1}, {2}, {3} };
    cout << "Median is " << binaryMedian(m, r, c) << endl;
    return 0;
}
```

input

```
Median is 2


...Program finished with exit code 0
Press ENTER to exit console.
```