Passing 2D arrays as arguments in C++

C++ does not allow to pass an entire array as an argument to a function. However, You can pass a pointer to an array by specifying the array's name without an index. There are three ways to pass a 2D array to a function −

Specify the size of columns of 2D array

```
void processArr(int a[][10]) {
   // Do something
}
```

Pass array containing pointers

```
void processArr(int *a[10]) {
   // Do Something
}

// When callingint *array[10];
for(int i = 0; i < 10; i++)
   array[i] = new int[10];
processArr(array);
```

Pass a pointer to a pointer

```
void processArr(int **a) {
   // Do Something
}
// When calling:
int **array;
array = new int *[10];
for(int i = 0; i <10; i++)
   array[i] = new int[10];
processArr(array);
```

EXAMPLE -

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

void processArr(int a[][2]) {
   cout << "element at index 1,1 is " << a[1][1];
}
int main() {
   int arr[2][2];
   arr[0][0] = 0;
   arr[0][1] = 1;
   arr[1][0] = 2;
   arr[1][1] = 3;

   processArr(arr);
   return 0;
}
```

Output

```
element at index 1,1 is 3
```