

```

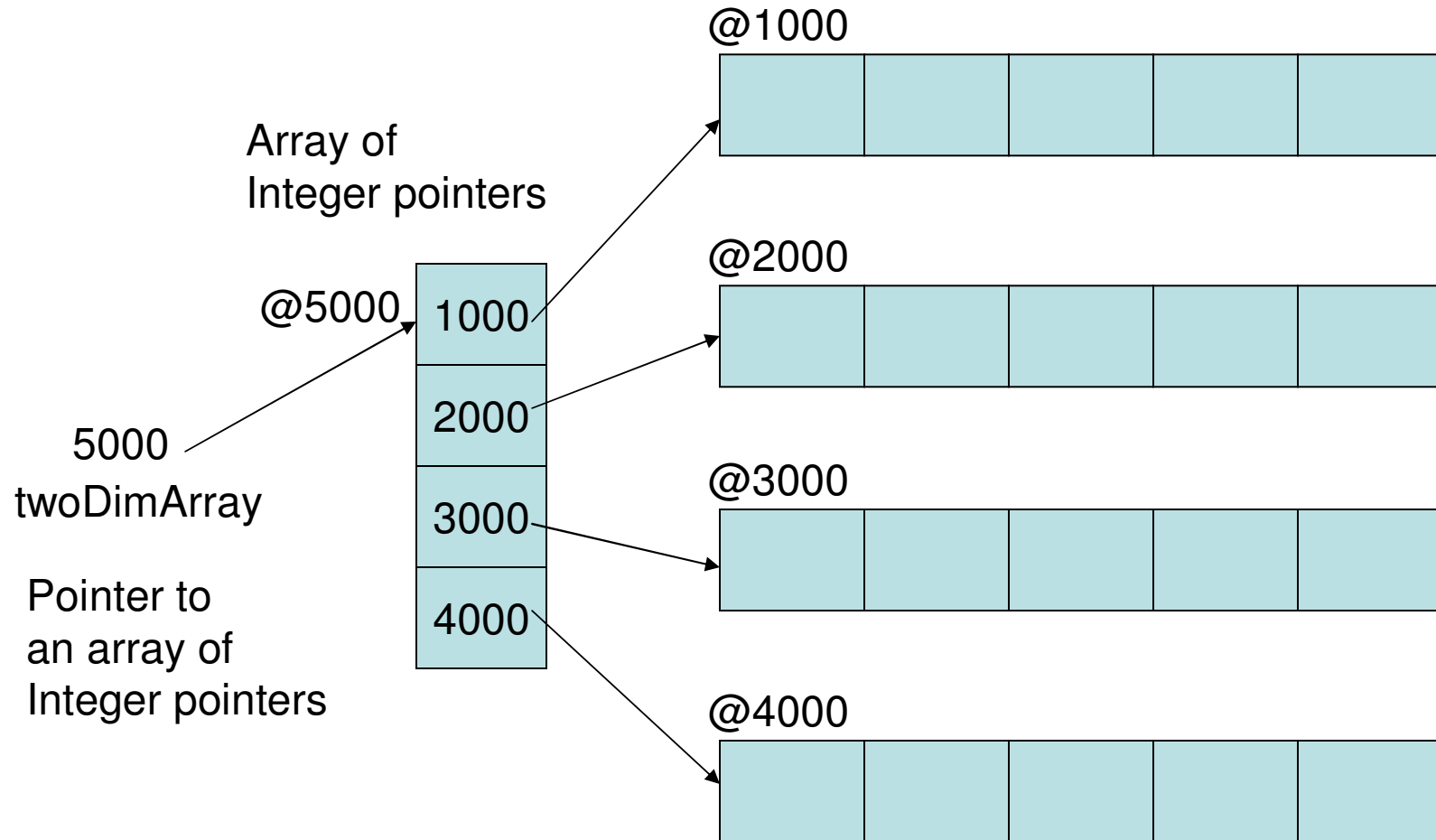
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6
7      int numRows;
8      int numColumns;
9
10     cout << "Enter the number of rows: ";
11     cin >> numRows;
12
13     cout << "Enter the number of columns: ";
14     cin >> numColumns;
15
16
17     int** twoDimArray = new int* [numRows];
18     // twoDimArray is a pointer to an array of integer pointers
19     // twoDimArray stores the base address of the array of integer pointers
20
21     for (int rowindex = 0; rowindex < numRows; rowindex++){
22         twoDimArray[rowindex] = new int[numColumns];
23         // each entry in the array of integer pointers stores the
24         // base address of an array of integers
25     }
26
27     int value = 1;
28
29     for (int rowindex = 0; rowindex < numRows; rowindex++){
30
31         for (int colindex = 0; colindex < numColumns; colindex++){
32
33             twoDimArray[rowindex][colindex] = value;
34             value++;
35         }
36
37     }
38
39
40     for (int rowindex = 0; rowindex < numRows; rowindex++){
41
42         for (int colindex = 0; colindex < numColumns; colindex++){
43
44             cout << twoDimArray[rowindex][colindex] << " ";
45         }
46
47         cout << endl;
48
49     }
50
51
52
53
54
55
56     system("pause");
57
58     return 0;
59
60 }

```

```

Enter the number of rows: 4
Enter the number of columns: 5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

```



`twoDimArray[2][3]` means we are accessing the cell at index 2 of the array of Integer pointers; the value in that cell is 3000 and since the cell is of type integer pointer, the 3000 is treated as the base address of an array of integers. We are then accessing the cell at index 3 of this array.