

```

1  #include <stdio.h>
2
3  #define NUM_ROWS 5
4  #define NUM_COLUMNS 3
5
6  int main(void)
7  {
8      //variable declarations
9      int iArray[NUM_ROWS][NUM_COLUMNS];
10     int i, j;
11
12     int *ptr_iArray_Row = NULL;
13
14     //code
15     // *** EVERY ROW OF A 2D ARRAY IS AN INTEGER ARRAY ITSELF COMPRISING OF
16     // *** 'NUM_COLUMNS' INTEGER ELEMENTS ***
17     // *** THERE ARE 5 ROWS AND 3 COLUMNS IN A 2D INTEGER ARRAY. EACH OF THE 5
18     // *** ROWS IS A 1D ARRAY OF 3 INTEGERS.
19     // *** HENCE, EACH OF THESE 5 ROWS THEMSELVES BEING ARRAYS, WILL BE THE
20     // *** BASE ADDRESSES OF THEIR RESPECTIVE ROWS ***
21
22     for (i = 0; i < NUM_ROWS; i++)
23     {
24         ptr_iArray_Row = iArray[i]; // 'iArray[i]' IS THE BASE ADDRESS OF ith
25         ROW ...
26         for (j = 0; j < NUM_COLUMNS; j++)
27             *(ptr_iArray_Row + j) = (i + 1) * (j + 1); //
28             'ptr_iArray_Row' (That is, 'iArray[i]' Can Be Treated As 1D Array
29             Using Pointers) ...
30     }
31
32     printf("\n\n");
33     printf("2D Integer Array Elements Along With Addresses : \n\n");
34     for (i = 0; i < NUM_ROWS; i++)
35     {
36         ptr_iArray_Row = iArray[i];
37         for (j = 0; j < NUM_COLUMNS; j++)
38         {
39             printf("(ptr_iArray_Row + %d)= %d \t \t At Address (ptr_iArray_Row
40             + j) : %p\n", j, *(ptr_iArray_Row + j), (ptr_iArray_Row + j));
41         }
42         printf("\n\n");
43     }
44     return(0);
45 }

```