

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define NUM_ROWS 5
5  #define NUM_COLUMNS 5
6
7  int main(void)
8  {
9      //variable declarations
10     int *iArray[NUM_ROWS]; //A 2D Array which will have 5 rows and number of
        columns can be decided later on ...
11     int i, j;
12
13     //code
14     printf("\n\n");
15     printf("***** MEMORY ALLOCATION TO 2D INTEGER ARRAY *****\n\n");
16     for (i = 0; i < NUM_ROWS; i++)
17     {
18         //ROW 0 WILL HAVE (NUM_COLUMNS - 0) = (5 - 0) = 5 COLUMNS...
19         //ROW 1 WILL HAVE (NUM_COLUMNS - 1) = (5 - 1) = 4 COLUMNS...
20         //ROW 2 WILL HAVE (NUM_COLUMNS - 2) = (5 - 2) = 3 COLUMNS...
21         //ROW 3 WILL HAVE (NUM_COLUMNS - 3) = (5 - 3) = 2 COLUMNS...
22         //ROW 4 WILL HAVE (NUM_COLUMNS - 4) = (5 - 4) = 1 COLUMN...
23
24         //BEACUSE OF THIS, THERE IS NO CONTIGUOUS MEMORY ALLOCATION ... HENCE,
        ALTHOUGH WE MAY USE THE DATA AS A 2D ARRAY, IT IS NOT REALLY A 2D
        ARRAY IN MEMORY ...
25
26         iArray[i] = (int *)malloc((NUM_COLUMNS - i) * sizeof(int));
27         if (iArray[i] == NULL)
28         {
29             printf("FAILED TO ALLOCATE MEMORY TO ROW %d OF 2D INTEGER ARRAY !!!
                EXITTING NOW...\n\n", i);
30             exit(0);
31         }
32         else
33             printf("MEMORY ALLOCATION TO ROW %d OF 2D INTEGER ARRAY
                SUCCEEDED !!!\n\n", i);
34     }
35
36     for (i = 0; i < 5; i++)
37     {
38         for (j = 0; j < (NUM_COLUMNS - i); j++)
39         {
40             iArray[i][j] = (i * 1) + (j * 1);
41         }
42     }
43
44     for (i = 0; i < 5; i++)
45     {
46         for (j = 0; j < (NUM_COLUMNS - i); j++)
47         {
48             printf("iArray[%d][%d] = %d \t At Address : %p\n", i, j, iArray[i]
                [j], &iArray[i][j]);
49         }
50         printf("\n");
    }
```

```
51     }
52
53     for (i = (NUM_ROWS - 1); i >= 0; i--)
54     {
55         if (iArray[i])
56         {
57             free(iArray[i]);
58             iArray[i] = NULL;
59             printf("MEMORY ALLOCATED TO ROW %d HAS BEEN SUCCESSFULLY FREED !!! \n\n", i);
60         }
61     }
62
63     return(0);
64 }
65
66
67
```