

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define NUM_ROWS 5
5  #define NUM_COLUMNS 3
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     for (i = 0; i < NUM_ROWS; i++)
16     {
17         iArray[i] = (int *)malloc(NUM_COLUMNS * sizeof(int));
18         if (iArray[i] == NULL)
19         {
20             printf("FAILED TO ALLOCATE MEMORY TO ROW %d OF 2D INTEGER ARRAY !!!
                EXITTING NOW...\n\n", i);
21             exit(0);
22         }
23         else
24             printf("MEMORY ALLOCATION TO ROW %d OF 2D INTEGER ARRAY
                SUCCEEDED !!!\n\n", i);
25     }
26
27     //ASSIGNING VALUES TO 2D ARRAY ...
28     for (i = 0; i < NUM_ROWS; i++)
29     {
30         for (j = 0; j < NUM_COLUMNS; j++)
31         {
32             iArray[i][j] = (i + 1) * (j + 1);
33         }
34     }
35
36     //DISPLAYING 2D ARRAY ...
37     printf("\n\n");
38     printf("DISPLAYING 2D ARRAY : \n\n");
39     for (i = 0; i < NUM_ROWS; i++)
40     {
41         for (j = 0; j < NUM_COLUMNS; j++)
42         {
43             printf("iArray[%d][%d] = %d\n", i, j, iArray[i][j]);
44         }
45         printf("\n\n");
46     }
47     printf("\n\n");
48
49     //FREEING MEMORY ASSIGNED TO 2D ARRAY (MUST BE DONE IN REVERSE ORDER)
50     for (i = (NUM_ROWS - 1); i >= 0; i--)
51     {
52         free(iArray[i]);
53         iArray[i] = NULL;
```

```
54     printf("MEMORY ALLOCATED TO ROW %d Of 2D INTEGER ARRAY HAS BEEN  
        SUCCESSFULLY FREED !!!\n\n", i);
```

```
55     }
```

```
56
```

```
57     return(0);
```

```
58 }
```

```
59
```

```
60
```

```
61
```