```c
#include <stdio.h>
#include <stdlib.h>

#define NUM_ROWS 5
#define NUM_COLUMNS_ONE 3
#define NUM_COLUMNS_TWO 8

int main(void)
{
    //variable declarations
    int *iArray[NUM_ROWS]; //A 2D Array which will have 5 rows and number of columns can be decided later on ...
    int i, j;

    //code

    // ********** ONE (ALLOCATING MEMORY FOR AN ARRAY OF 3 INTEGERS PER ROW) **********
    printf("\n\n");
    printf("********** FIRST MEMORY ALLOCATION TO 2D INTEGER ARRAY **********\n\n");
    for (i = 0; i < NUM_ROWS; i++)
    {
        iArray[i] = (int *)malloc(NUM_COLUMNS_ONE * sizeof(int));
        if (iArray[i] == NULL)
        {
            printf("FAILED TO ALLOCATE MEMORY TO ROW %d OF 2D INTEGER ARRAY !!! EXITTING NOW...\n\n", i);
            exit(0);
        }
        else
            printf("MEMORY ALLOCATION TO ROW %d OF 2D INTEGER ARRAY SUCCEEDED !!!\n\n", i);
    }

    //ASSIGNING VALUES TO 2D ARRAY ...
    for (i = 0; i < NUM_ROWS; i++)
    {
        for (j = 0; j < NUM_COLUMNS_ONE; j++)
        {
            iArray[i][j] = (i + 1) * (j + 1);
        }
    }

    //DISPLAYING 2D ARRAY ...
    printf("\n\n");
    printf("DISPLAYING 2D ARRAY : \n\n");
    for (i = 0; i < NUM_ROWS; i++)
    {
        for (j = 0; j < NUM_COLUMNS_ONE; j++)
        {
            printf("iArray[%d][%d] = %d\n", i, j, iArray[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");

    //FREEING MEMORY ASSIGNED TO 2D ARRAY (MUST BE DONE IN REVERSE ORDER)
    for (i = (NUM_ROWS - 1); i >= 0; i--)
    {
        free(iArray[i]);
        iArray[i] = NULL;
        printf("MEMORY ALLOCATED TO ROW %d Of 2D INTEGER ARRAY HAS BEEN SUCCESSFULLY FREED !!!\n\n", i);
    }

    // ********** TWO (ALLOCATING MEMORY FOR AN ARRAY OF 8 INTEGERS PER ROW) **********
    printf("\n\n");
    printf("********** SECOND MEMORY ALLOCATION TO 2D INTEGER ARRAY **********\n\n");
    for (i = 0; i < NUM_ROWS; i++)
    {
        iArray[i] = (int *)malloc(NUM_COLUMNS_TWO * sizeof(int));
        if (iArray[i] == NULL)
        {
            printf("FAILED TO ALLOCATE MEMORY TO ROW %d OF 2D INTEGER ARRAY !!! EXITTING NOW...\n\n", i);
            exit(0);
        }
        else
            printf("MEMORY ALLOCATION TO ROW %d OF 2D INTEGER ARRAY SUCCEEDED !!!\n\n", i);
    }

    //ASSIGNING VALUES TO 2D ARRAY ...
    for (i = 0; i < NUM_ROWS; i++)
    {
        for (j = 0; j < NUM_COLUMNS_TWO; j++)
        {
            iArray[i][j] = (i + 1) * (j + 1);
        }
    }

    //DISPLAYING 2D ARRAY ...
    printf("\n\n");
    printf("DISPLAYING 2D ARRAY : \n\n");
    for (i = 0; i < NUM_ROWS; i++)
    {
        for (j = 0; j < NUM_COLUMNS_TWO; j++)
        {
            printf("iArray[%d][%d] = %d\n", i, j, iArray[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");

    //FREEING MEMORY ASSIGNED TO 2D ARRAY (MUST BE DONE IN REVERSE ORDER)
    for (i = (NUM_ROWS - 1); i >= 0; i--)
    {
        free(iArray[i]);
        iArray[i] = NULL;
        printf("MEMORY ALLOCATED TO ROW %d Of 2D INTEGER ARRAY HAS BEEN 
          SUCCESSFULLY FREED !!!\n\n", i);
    }

    return(0);
}
```