**Code for traversal of 2D Array**

|  |
| --- |
| #include <stdio.h>    #define N 2  #define M 3    // Function to traverse the matrix recursively  int traverseMatrix(int arr[N][M], int current\_row,  int current\_col)  {  // If the entire column is traversed  if (current\_col >= M)  return 0;    // If the entire row is traversed  if (current\_row >= N)  return 1;    // Print the value of the current  // cell of the matrix  printf("%d, ", arr[current\_row][current\_col]);;    // Recursive call to traverse the matrix  // in the Horizontal direction  if (traverseMatrix(arr, current\_row,  current\_col + 1)  == 1)  return 1;    // Recursive call for changing the  // Row of the matrix  return traverseMatrix(arr,  current\_row + 1,  0);  }    // Driver code  int main()  {  int arr[N][M] = { { 1, 2, 3 },  { 4, 5, 6 } };    traverseMatrix(arr, 0, 0);  return 0;  } |

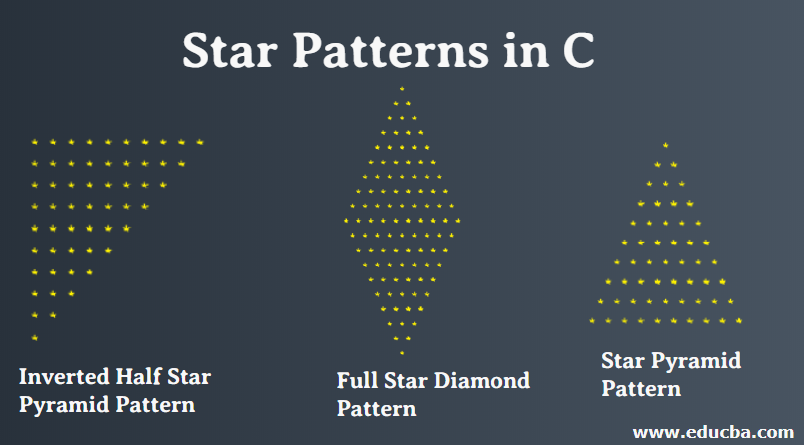
**Exercise # 10**

1. Continue working on tasks of previous lab. Make a clearer understanding of functions.
2. Create a C-program to perform multiplication of two matrices, but the dimensions of matrices should be provided by the user.
3. Write a program to calculate the factorial of any number using recursion, also draw the stack diagram for recursion.
4. Write a C-program to revert a decimal string using recursion and determine whether its palindrome or not.
5. Write a C-program using recursion which calculates the number of 1s in 2D array. The size of 2D array must be provided by the user.

Here is the count is number 1s in the array.

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

1. Write a C program to draw the following pattern using recursion.



1. Write a C program to multiply two matrices using recursion.
2. Write a C program to find a min and max element in array using recursion.