```
/*

Que : 1.Write C Program to read and print a Matrix , R and C must be input by User.
(Using Dynamic Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.
#include<stdlib.h>

void main() {
	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}
}




/*
```

```
Que : 2.Write a C Program to Search Element in a 2D Array (Using Dynamic Memory
Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        ********  Solution  ********


#include<stdio.h> //Include Necessary Header Files.

void main() {
	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}

	// Logic to search Element in 2D array using DMA.
	int search, flag = 0;
	printf("Enter the Search Element: ");
	scanf_s("%d", &search);

	for (int i = 0; i < row; i++) // For loop to search element in array.
	{
		for (int j = 0; j < col; j++)
		{
```

```
                        if (*(*(p + i) + j) == search) // If element found then break the
loop..
                        {
                                flag = 1;
                                break;
                        }
                }
                if (flag == 1) // If element found then break the loop..
                {
                        break;
                }
        }
        if (flag == 1) {
                printf("The given element %d is Present in array..", search);
        }
        else {
                printf("The given element %d is Not Present in array..", search);
        }
}




/*

Que : 3.Write a C Program to find the transpose of a given matrix (Using Dynamic
Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.

void main() {
        int row, col; // Defination of required variables

        printf("Enter Number Of Rows:");
        scanf_s("%d", &row); // Take input - Number of array elements in row.

        printf("Enter Number of Columns:");
        scanf_s("%d", &col); // Take input - Number of array elements in columns.

        int** p = NULL; // Defination of pointer array..

        p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

        for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
        {
                *(p + i) = (int*)malloc(col * sizeof(int));
        }

        printf("Enter array Elements...\n");

        for (int i = 0; i < row; i++) // For loop to take input array elements.
        {
                for (int j = 0; j < col; j++)
```

```
			{
				scanf_s("%d", *(p + i) + j);
			}
		}

		printf("Array Elements are: \n");
		for (int i = 0; i < row; i++) // For loop to print array elements.
		{
			for (int j = 0; j < col; j++)
			{
				printf("%d\t", *(*(p + i) + j));
			}
			printf("\n");
		}

		// Logic for transpose of a matrix.
		printf("The Transpose of given matrix is:\n");
		for (int i = 0; i < row; i++) // For loop to print transpose of matrix..
		{
			for (int j = 0; j < col; j++)
			{
				printf("%d\t", *(*(p + j) + i));
			}
			printf("\n");
		}

}




/*

Que : 4. Write a C program to add two matrices in third Matrix(Using Dynamic Memory
Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.
#include<stdlib.h>

void main() {
		int row, col; // Defination of required variables

		printf("Enter Number Of Rows:");
		scanf_s("%d", &row); // Take input - Number of array elements in row.

		printf("Enter Number of Columns:");
		scanf_s("%d", &col); // Take input - Number of array elements in columns.

		// Array 1 creation logic..
		int** p = NULL; // Defination of pointer array..
		p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...
		for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
		{
```

```
			*(p + i) = (int*)malloc(col * sizeof(int));
		}

		// Array 2 creation logic..
		int** q = NULL; // Defination of pointer array..
		q = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer q
dynamically...
		for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
		{
			*(q + i) = (int*)malloc(col * sizeof(int));
		}

		// Array 3 for storing the result of addition of array 1 and 2 ...
		int** r = NULL; // Defination of pointer array..
		r = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer r
dynamically...
		for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
		{
			*(r + i) = (int*)malloc(col * sizeof(int));
		}


		printf("Enter array Elements of first matrix...\n");
		for (int i = 0; i < row; i++) // For loop to take input array elements.
		{
			for (int j = 0; j < col; j++)
			{
				scanf_s("%d", *(p + i) + j);
			}
		}

		printf("Enter array Elements of second matrix...\n");
		for (int i = 0; i < row; i++) // For loop to take input array elements.
		{
			for (int j = 0; j < col; j++)
			{
				scanf_s("%d", *(q + i) + j);
			}
		}

		printf("First array is: \n");
		for (int i = 0; i < row; i++) // For loop to print array elements.
		{
			for (int j = 0; j < col; j++)
			{
				printf("%d\t", *(*(p + i) + j));
			}
			printf("\n");
		}

		printf("Second array is: \n");
		for (int i = 0; i < row; i++) // For loop to print array elements.
		{
			for (int j = 0; j < col; j++)
			{
				printf("%d\t", *(*(q + i) + j));
			}
			printf("\n");
		}
```

```
	// Logic for addition of two matrices...
	printf("The addition of given two matrices is:\n");
	for (int i = 0; i < row; i++) // For loop to add two matrices..
	{
		for (int j = 0; j < col; j++)
		{
			*(*(r + i) + j) = *(*(p + i) + j) + *(*(q + i) + j);
		}
	}

	for (int i = 0; i < row; i++) // For loop to print addition matrix.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(r + i) + j));
		}
		printf("\n");
	}

}




/*

Que : 5. Write a C program to substract two matrices in third Matrix(Using Dynamic
Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.
#include<stdlib.h>

void main() {
	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	// Array 1 creation logic..
	int** p = NULL; // Defination of pointer array..
	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...
	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	// Array 2 creation logic..
	int** q = NULL; // Defination of pointer array..
	q = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer q
dynamically...
```

```
	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D pointer array
	{
		*(q + i) = (int*)malloc(col * sizeof(int));
	}

	// Array 3 for storing the result of substraction of array 1 and 2 ...
	int** r = NULL; // Defination of pointer array..
	r = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer r dynamically...
	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D pointer array
	{
		*(r + i) = (int*)malloc(col * sizeof(int));
	}


	printf("Enter array Elements of first matrix...\n");
	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Enter array Elements of second matrix...\n");
	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(q + i) + j);
		}
	}

	printf("First array is: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}

	printf("Second array is: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(q + i) + j));
		}
		printf("\n");
	}

	// Logic for Substraction of two matrices...
	printf("The substraction of given two matrices is:\n");
	for (int i = 0; i < row; i++) // For loop to add two matrices..
	{
		for (int j = 0; j < col; j++)
		{
			*(*(r + i) + j) = *(*(p + i) + j) - *(*(q + i) + j);
```

```
			}
		}

		for (int i = 0; i < row; i++) // For loop to print substraction matrix.
		{
			for (int j = 0; j < col; j++)
			{
				printf("%d\t", *(*(r + i) + j));
			}
			printf("\n");
		}

}




/*

Que : 6. Write a c program to check whether given matrix is upper triangular or not
(Using Synamic Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.

void main() {
		int row, col; // Defination of required variables

		printf("Enter Number Of Rows:");
		scanf_s("%d", &row); // Take input - Number of array elements in row.

		printf("Enter Number of Columns:");
		scanf_s("%d", &col); // Take input - Number of array elements in columns.

		int** p = NULL; // Defination of pointer array..

		p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

		for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
		{
			*(p + i) = (int*)malloc(col * sizeof(int));
		}

		printf("Enter array Elements...\n");

		for (int i = 0; i < row; i++) // For loop to take input array elements.
		{
			for (int j = 0; j < col; j++)
			{
				scanf_s("%d", *(p + i) + j);
			}
		}

		printf("Array Elements are: \n");
```

```
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}

	// Logic to check given matrix is upper triangular or not..
	int flag = 0;
	for (int i = 1; i < row; i++) {
		for (int j = 0; j < i; j++) {
			if (*(*(p + i) + j) != 0) {
				flag = 1;
				break;
			}
		}
		if (flag == 1) {
			break;
		}
	}

	if (flag == 1) {
		printf("The given matrix is not an upper triangular matrix.");
	}
	else {
		printf("The given matrix is an upper triangular matrix.");
	}
}




/*

Que : 7. Write a C program to check whether given matrix is lower triangular or not
(Using Dynamic Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.

void main() {
	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...
```

```
	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}


	// Logic to check given matrix is lower triangular or not..
	int flag = 0;
	for (int i = 1; i < row; i++) {
		for (int j = 0; j < i; j++) {
			if (*(*(p + j) + i) != 0) {
				flag = 1;
				break;
			}
		}
		if (flag == 1) {
			break;
		}
	}

	if (flag == 1) {
		printf("The given matrix is not an lower triangular matrix.");
	}
	else {
		printf("The given matrix is an lower triangular matrix.");
	}
}




/*

Que : 8. Write C Program to Check if a given Matrix is an Unit Matrix. (Using Dynamic Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                      *********  Solution  *********
```

```c
#include<stdio.h> //Include Necessary Header Files.

void main() {

	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}


	// Logic to check given matrix is unit matrix or not..
	int flag = 0;
	for (int i = 0; i < row; i++)
	{
		for (int j = 0; j < col; j++)
		{
			if (*(*(p + i) + j) != 1) {
				flag = 1;
				break;
			}
		}
		if (flag == 1) {
			break;
		}
	}
```

```
	if (flag == 1) {
		printf("The given matrix is not an unit matrix.");
	}
	else {
		printf("The given matrix is an unit matrix.");
	}

}




/*

Que : 9. Write a C Program to check whether a given matrix is an identity matrix or
not (Using Dynamic Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                        *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.
#include<stdlib.h>

void main() {
	int row, col; // Defination of required variables

	printf("Enter Number Of Rows:");
	scanf_s("%d", &row); // Take input - Number of array elements in row.

	printf("Enter Number of Columns:");
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
```

```
                printf("%d\t", *(*(p + i) + j));
            }
            printf("\n");
        }


        // Logic to check given matrix is Identity matrix or not..
        int flag = 0;
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                if (i == j) {
                    if (*(*(p + i) + j) != 1) {
                        flag = 1;
                        break;
                    }
                }
                else {
                    if (*(*(p + i) + j) != 0) {
                        flag = 1;
                        break;
                    }
                }
            }
            if (flag == 1) {
                break;
            }
        }

        if (flag == 1) {
            printf("The given matrix is not an Identity matrix.");
        }
        else {
            printf("The given matrix is an Identity matrix.");
        }
}




/*

Que : 10. Write C program to check if the matrix is symmetric or not (Using Dynamic
Memory Allocation).
Owner: Rushikesh Sanjay Pokharkar
Batch: PPA9

*/

//                                      *********  Solution  *********


#include<stdio.h> //Include Necessary Header Files.

void main() {
        int row, col; // Defination of required variables

        printf("Enter Number Of Rows:");
        scanf_s("%d", &row); // Take input - Number of array elements in row.

        printf("Enter Number of Columns:");
```

```
	scanf_s("%d", &col); // Take input - Number of array elements in columns.

	int** p = NULL; // Defination of pointer array..

	p = (int**)malloc(row * sizeof(int*)); // Allocating memory to 2D pointer p
dynamically...

	for (int i = 0; i < col; i++) // For loop to assigne address of 1D array in 2D
pointer array
	{
		*(p + i) = (int*)malloc(col * sizeof(int));
	}

	printf("Enter array Elements...\n");

	for (int i = 0; i < row; i++) // For loop to take input array elements.
	{
		for (int j = 0; j < col; j++)
		{
			scanf_s("%d", *(p + i) + j);
		}
	}

	printf("Array Elements are: \n");
	for (int i = 0; i < row; i++) // For loop to print array elements.
	{
		for (int j = 0; j < col; j++)
		{
			printf("%d\t", *(*(p + i) + j));
		}
		printf("\n");
	}


	// Logic to check given matrix is Symmitric matrix or not..
	int flag = 0;
	for (int i = 0; i < row; i++)
	{
		for (int j = 0; j < col; j++)
		{
			if (*(*(p + i) + j) != *(*(p + j) + i)) {
				flag = 1;
				break;
			}
		}
		if (flag == 1) {
			break;
		}
	}
	if (flag == 1) {
		printf("The given matrix is not an symmitric matrix");
	}
	else {
		printf("The given matrix is an symmitric matrix");
	}
}
```