```c
/*
* Que.1 : User define 2D matrix and print as it is using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int **p = NULL;
       int i , j;
       int row , col;

       printf("how many rows you want : ");
       scanf("%d",&row);
       printf("how many columns you want : ");
       scanf("%d",&col);

       p = (int**)malloc(row*sizeof(int*));

       for(i = 0 ; i < row ; i++)
       {
              *(p+i) = (int*)malloc(row*sizeof(int));
       }

       printf("enter values : \n");

       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     scanf("%d",*(p+i)+j);
              }
       }

       printf("array values : \n");
       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     printf("%d\t",*(*(p+i)+j));
              }
              printf("\n");
       }

       getch();
}
```

```c
/*
* Que.2 : Search element in user define 2D matrix using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
	int **p = NULL;
	int i , j;
	int row , col;
	int search;
	int f = 0;

	printf("how many rows you want : ");
	scanf("%d",&row);
	printf("how many columns you want : ");
	scanf("%d",&col);

	p = (int**)malloc(row*sizeof(int*));

	for(i = 0 ; i < row ; i++)
	{
		*(p+i) = (int*)malloc(row*sizeof(int));
	}

	printf("enter values : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			scanf("%d",*(p+i)+j);
		}
	}

	printf("enter element you want to search : ");
	scanf("%d",&search);

	printf("array values : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			printf("%d\t",*(*(p+i)+j));
			if(search == *(*(p+i)+j))
			{
				f = 1;
			}
		}
		printf("\n");
	}
```

```
    if(f == 1)
        printf("element is found !!");
    else
        printf("element is not found !!");

    getch();
}
```

```c
/*
* Que.3 : Find transpose of user define 2D matrix using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
	int **p = NULL;
	int **q = NULL;
	int i , j;
	int row , col;

	printf("how many rows you want : ");
	scanf("%d",&row);
	printf("how many columns you want : ");
	scanf("%d",&col);

	p = (int**)malloc(row*sizeof(int*));

	for(i = 0 ; i < row ; i++)
	{
		*(p+i) = (int*)malloc(row*sizeof(int));
	}

	printf("enter values : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			scanf("%d",*(p+i)+j);
		}
	}

	printf("array values : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			printf("%d\t",*(*(p+i)+j));
		}
		printf("\n");
	}

	q = (int**)malloc(row*sizeof(int*));

	for(i = 0 ; i < row ; i++)
	{
		*(q+i) = (int*)malloc(row*sizeof(int));
	}
```

```
    printf("transpose array : \n");
    for(i = 0 ; i < row ; i++)
    {
        for(j = 0 ; j < col ; j++)
        {
            *(*(q+i)+j) = *(*(p+i)+j);
        }
    }

    for(i = 0 ; i < col ; i++)
    {
        for(j = 0 ; j < row ; j++)
        {
            printf("%d\t",*(*(q+j)+i));
        }
        printf("\n");
    }

    getch();
}
```

```
/*
* Que.4 : Find addition of user define 2D matrices using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int **p = NULL;
        int **q = NULL;
        int **r = NULL;
        int i , j;
        int row , col;

        printf("how many rows you want : ");
        scanf("%d",&row);
        printf("how many columns you want : ");
        scanf("%d",&col);

        p = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(p+i) = (int*)malloc(row*sizeof(int));
        }

        printf("enter values for 1st matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(p+i)+j);
                }
        }

        printf("array values in 1st matrix: \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(p+i)+j));
                }
                printf("\n");
        }

        q = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(q+i) = (int*)malloc(row*sizeof(int));
        }
        printf("enter values for 2nd matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(q+i)+j);
                }
        }

        printf("array values in 2nd matrix: \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(q+i)+j));
                }
                printf("\n");
        }

        r = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(r+i) = (int*)malloc(row*sizeof(int));
        }

        printf("addition of 2 matrices is : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        *(*(r+i)+j) = *(*(p+i)+j) + *(*(q+i)+j);
                        printf("%d\t",*(*(r+i)+j));
                }
                printf("\n");
        }

        getch();
}
```

```c
/*
* Que.5 : Find subtraction of user define 2D matrices using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int **p = NULL;
        int **q = NULL;
        int **r = NULL;
        int i , j;
        int row , col;

        printf("how many rows you want : ");
        scanf("%d",&row);
        printf("how many columns you want : ");
        scanf("%d",&col);

        p = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(p+i) = (int*)malloc(row*sizeof(int));
        }

        printf("enter values for 1st matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(p+i)+j);
                }
        }

        printf("array values in 1st matrix: \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(p+i)+j));
                }
                printf("\n");
        }

        q = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(q+i) = (int*)malloc(row*sizeof(int));
        }
        printf("enter values for 2nd matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(q+i)+j);
                }
        }

        printf("array values in 2nd matrix: \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(q+i)+j));
                }
                printf("\n");
        }

        r = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(r+i) = (int*)malloc(row*sizeof(int));
        }

        printf("subtraction of 2 matrices is : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        *(*(r+i)+j) = *(*(p+i)+j) - *(*(q+i)+j);
                        printf("%d\t",*(*(r+i)+j));
                }
                printf("\n");
        }

        getch();
}
```

```
/*
* Que.6 : Check given 2D matrix is upper triangular or not using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
	int **p = NULL;
	int i , j;
	int row , col;
	int cnt1 = 0 , cnt2 = 0;

	printf("how many rows you want : ");
	scanf("%d",&row);
	printf("how many columns you want : ");
	scanf("%d",&col);

	p = (int**)malloc(row*sizeof(int*));

	for(i = 0 ; i < row ; i++)
	{
		*(p+i) = (int*)malloc(row*sizeof(int));
	}

	printf("enter values in matrix : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			scanf("%d",*(p+i)+j);
		}
	}

	printf("matrix array values : \n");
	for(i = 0 ; i < row ; i++)
	{
		for(j = 0 ; j < col ; j++)
		{
			printf("%d\t",*(*(p+i)+j));
		}
		printf("\n");
	}

	if(row == col)
	{
		for(i = 0 ; i < row ; i++)
		{
			for(j = 0 ; j < i ; j++)
			{
				if(*(*(p+i)+j) == 0)
```

```
                {
                    cnt1++;
                }
                cnt2++;
            }
        }

        if(cnt1 == cnt2)
            printf("given matrix is upper triangular matrix");
        else
            printf("given matrix is not upper triangular matrix");
    }
    else
        printf("given matrix is not upper triangular matrix");


    getch();
}
```

```
/*
* Que.7 : Check given 2D matrix is lower triangular or not using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int **p = NULL;
       int i , j;
       int row , col;
       int cnt1 = 0 , cnt2 = 0;

       printf("how many rows you want : ");
       scanf("%d",&row);
       printf("how many columns you want : ");
       scanf("%d",&col);

       p = (int**)malloc(row*sizeof(int*));

       for(i = 0 ; i < row ; i++)
       {
              *(p+i) = (int*)malloc(row*sizeof(int));
       }

       printf("enter values in matrix : \n");
       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     scanf("%d",*(p+i)+j);
              }
       }

       printf("matrix array values : \n");
       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     printf("%d\t",*(*(p+i)+j));
              }
              printf("\n");
       }

       if(row == col)
       {
              for(i = 0 ; i < row ; i++)
              {
                     for(j = i+1 ; j < col ; j++)
                     {
                            if(*(*(p+i)+j) == 0)
```

```
                {
                    cnt1++;
                }
                cnt2++;
            }
        }

        if(cnt1 == cnt2)
            printf("given matrix is lower triangular matrix");
        else
            printf("given matrix is not lower triangular matrix");
    }
    else
        printf("given matrix is not lower triangular matrix");


    getch();
}
```

```c
/*
* Que.8 : Check given 2D matrix is unit matrix or not using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int **p = NULL;
       int i , j;
       int row , col;
       int cnt = 0 , cnt1 = 0;

       printf("how many rows you want : ");
       scanf("%d",&row);
       printf("how many columns you want : ");
       scanf("%d",&col);

       cnt = row*col;

       p = (int**)malloc(row*sizeof(int*));

       for(i = 0 ; i < row ; i++)
       {
              *(p+i) = (int*)malloc(row*sizeof(int));
       }

       printf("enter values in matrix : \n");
       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     scanf("%d",*(p+i)+j);
              }
       }

       printf("matrix array values : \n");
       for(i = 0 ; i < row ; i++)
       {
              for(j = 0 ; j < col ; j++)
              {
                     printf("%d\t",*(*(p+i)+j));
                     if(*(*(p+i)+j) == 1)
                     {
                            cnt1++;
                     }
              }
              printf("\n");
       }

       if(cnt == cnt1)
```

```
        printf("given matrix is unit matrix");
    else
        printf("given matrix is not unit matrix");

    getch();
}
```

```c
/*
* Que.9 : Check given 2D matrix is identity matrix or not using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int **p = NULL;
        int i , j;
        int row , col;
        int cnt = 0 , cnt1 = 0 , cnt2 = 0;
        int min;

        printf("how many rows you want : ");
        scanf("%d",&row);
        printf("how many columns you want : ");
        scanf("%d",&col);

        cnt = row*col;

        if(row >= col)
                min = col;
        else
                min = row;

        p = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(p+i) = (int*)malloc(row*sizeof(int));
        }

        printf("enter values in matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(p+i)+j);
                }
        }

        printf("matrix array values : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(p+i)+j));
                        if(i == j && *(*(p+i)+j) == 1)
                        {
                                cnt1++;
```

```
            }
            else if(*(*(p+i)+j) == 0)
            {
                cnt2++;
            }
        }
        printf("\n");
    }

    if(cnt == cnt1 + cnt2 && cnt1 == min)
        printf("given matrix is identity matrix");
    else
        printf("given matrix is not identity matrix");

    getch();
}
```

```c
/*
* Que.10: Check given 2D matrix is symmetric matrix or not using DMA
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int **p = NULL;
        int i , j;
        int row , col;
        int cnt = 0;

        printf("how many rows you want : ");
        scanf("%d",&row);
        printf("how many columns you want : ");
        scanf("%d",&col);

        p = (int**)malloc(row*sizeof(int*));

        for(i = 0 ; i < row ; i++)
        {
                *(p+i) = (int*)malloc(row*sizeof(int));
        }

        printf("enter values in matrix : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        scanf("%d",*(p+i)+j);
                }
        }

        printf("matrix array values : \n");
        for(i = 0 ; i < row ; i++)
        {
                for(j = 0 ; j < col ; j++)
                {
                        printf("%d\t",*(*(p+i)+j));
                }
                printf("\n");
        }

        if(row == col)
        {
                for(i = 0 ; i < row ; i++)
                {
                        for(j = 0 ; j < col ; j++)
                        {
                                if(*(*(p+i)+j) == *(*(p+j)+i) && i != j)
                {
                    cnt++;
                }
            }
        }

    if(cnt == row * col - row)
        printf("given matrix is symmetric matrix");
    else
        printf("given matrix is not symmetric matrix");
    }
    else
        printf("given matrix is not symmetric matrix");

    getch();
}
```