

1. WAP to input a 2D array of size M*N and display it in tabular form.

```
#include<stdio.h>

int main()
{
    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];

    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            printf("%d\t",a[i][j]);

        printf("\n");
    }

    return 0;
}
```

2. WAP to input a 2D array of size M*N and find the sum and average of all the elements.

```
#include<stdio.h>

int main()
{
    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];int s=0;float avg;
```

```

for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        scanf("%d",&a[i][j]);
        s=s+a[i][j];
    }
    avg=(float)s/(m*n);
    printf("Sum=%d\n Average=%f",s,avg);
    return 0;
}

```

3. WAP to input a 2D array of size M*N and find the sum of individual rows and individual columns.

```

#include<stdio.h>

```

```

int main()

```

```

{

```

```

    int m,n,i,j;

```

```

    scanf("%d%d",&m,&n);

```

```

    int a[m][n];int s=0;float avg;

```

```

for(i=0;i<m;i++)

```

```

    for(j=0;j<n;j++)

```

```

        scanf("%d",&a[i][j]);

```

```

for(i=0;i<m;i++)

```

```

{

```

```

    for(j=0;j<n;j++)

```

```

        s=s+a[i][j];

```

```

        printf("sum of row%d=%d\n",i+1,s);

        s=0;
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)

            s=s+a[j][i];

        printf("sum of column%d=%d\n",i+1,s);

        s=0;
    }

    return 0;
}

```

4. WAP to input a 2D array of size M*N and display boundary elements in matrix form.

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

```
int main()
```

```
{
```

```
    int m,n,i,j;
```

```
    scanf("%d%d",&m,&n);
```

```
    int a[m][n];
```

```
    for(i=0;i<m;i++)
```

```
    {
```

```
        for(j=0;j<n;j++)
```

```
        {
```

```
            scanf("%d",&a[i][j]);
```

```
        }
```

```

    }

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i==0 || i==m-1 || j==0 || j==n-1)
                printf("%d\t",a[i][j]);
            else printf("\t");
        }printf("\n");
    }

    return 0;
}

```

5. WAP to input a 2D array of size M*N and display the transpose of it.

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

```
int main()
```

```

{
    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}

```

```

    }

    for(i=0;i<n;i++)

    {

        for(j=0;j<m;j++)

            { printf("%d ",a[j][i]);

        }

        printf("\n");
    }

    return 0;

}

```

6. WAP to input a 2D array and find the sum of its diagonal elements.

```

#include<stdio.h>

int main()

{

    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];int s=0;

    if(m==n)

    {

        for(i=0;i<m;i++)

        {

            for(j=0;j<n;j++)

            {

                scanf("%d",&a[i][j]);

                if(i==j)s=s+a[i][j];
            }
        }
    }
}

```

```

        else if(i+j==n-1)s=s+a[i][j];

    }

}

printf("sum of diagonal elements = %d",s);

return 0;

}

```

7. WAP to input a 2D array and display diagonal elements in matrix form.

```

#include<stdio.h>

int main()

{

    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];

    if(m==n)

    {

        for(i=0;i<m;i++)

        {

            for(j=0;j<n;j++)

            {

                scanf("%d",&a[i][j]);

            }

        }

        for(i=0;i<m;i++)

```

```

{
    for(j=0;j<n;j++)
    {
        if(i==j)printf("%d ",a[i][j]);
        else if(i+j==n-1)printf("%d ",a[i][j]);
        else printf(" ");
    }printf("\n");
}
}
else printf("Invalid Input");

return 0;

}

```

8. WAP to input 2 matrices from the user and add them.

```

#include<stdio.h>

int main()
{
    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];int b[m][n];

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d%d",&a[i][j],&b[i][j]);
        }
    }
}

```

```

    }

    for(i=0;i<m;i++)

    {

        for(j=0;j<n;j++)

        { printf("%d ",a[i][j]+b[i][j]);

        }

        printf("\n");

    }

    return 0;

}

```

9. WAP to input a matrix and check if its identity matrix or not.

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

```
int main()
```

```

{

    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];int f=0;

    if(m==n)

    {

        for(i=0;i<m;i++)

        {

            for(j=0;j<n;j++)

            {

                scanf("%d",&a[i][j]);

                if(i==j&& a[i][j]!=1)f=1;

            }

        }

    }

}

```



```

        else if(i!=j&& a[i][j]!=0)f=1;

    }

}

if(f==1)printf("not identity matrix");

else printf("Identity matrix");

}

else printf("Matirx Must be square matrix");

return 0;

}

```

10. WAP to input a matrix of order M*N and check if it's sparse or dense matrix.

```

#include<stdio.h>

int main()

{

    int m,n,i,j;int z=0,d=0;

    scanf("%d%d",&m,&n);

    int a[m][n];int sumr=0,suml=0;

    for(i=0;i<m;i++)

    {

        for(j=0;j<n;j++)

        {

            scanf("%d",&a[i][j]);

            if(a[i][j]==0)z++;

            else d++;

        }

    }

```

```
}
```

```
if(z>d)printf("sparse");
```

```
else if(z<d)printf("Dense");
```

```
else printf("neither sparse nor dense");
```

```
return 0;
```

```
}
```

11. Write a program in C to calculate determinant of a 3 x 3 matrix.

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

```
int main()
```

```
{
```

```
int i,j,s=0;
```

```
int a[3][3];
```

```
for(i=0;i<3;i++)
```

```
for(j=0;j<3;j++)
```

```
scanf("%d",&a[i][j]);
```

```
for(i=0;i<3;i++)
```

```
s=s+a[0][i]*(a[1][(i+1)%3]*a[2][(i+2)%3]-a[1][(i+2)%3]*a[2][(i+1)%3]);
```

```
printf("Determinant = %d",s);
```

```
return 0;
```

```
}
```

12. Write a program to keep records and perform statistical analysis for a class of students. The class may have up to 10 students. There are three quizzes during the term for computer programming. Each student is identified by a four digit roll no. The program will print the student scores and calculate and print the following statistics

1. High score & Low score for each quiz along with the roll number of that student.
2. Average of Each Quiz and overall average of all the 3 quizzes

3. Highest Average and Lowest average of all the three quizzes along with quiz no.
4. Highest marks & lowest marks of each student in the 3 quizzes along with the quiz no.

```
#include<stdio.h>

int main()
{
    int i,j;printf("Enter no. of students\n");

    int n;scanf("%d",&n);

    int a[n];int b[3][n];

    printf("Enter roll. nos. of students\n");

    for(i=0;i<n;i++)

        scanf("%d",&a[i]);

    printf("Enter scores of students\n");

    for(i=0;i<3;i++)

        {printf("Quiz %d => ",i+1);

            for(j=0;j<n;j++)

                {

                    scanf("%d",&b[i][j]);

                }

        }

    float avg[3];

    for(i=0;i<3;i++)

        {printf("Quiz %d => ",i+1);int s=b[i][0],max=b[i][0],min=b[i][0];

            for(j=1;j<n;j++)

                {s=s+b[i][j];

                    if(b[i][j]>max)

                        max=b[i][j];
```

```

        if(b[i][j]<min)
            min=b[i][j];
    }
    printf("High score=Roll No.");

    for(j=0;j<n;j++)

        if(b[i][j]==max)printf("%d",a[j]);

    printf("\b=%d\tLow score=Roll no.",max);

    for(j=0;j<n;j++)

        if(b[i][j]==min)printf("%d",a[j]);

    printf("\b=%d\n",min);

    avg[i]=(float)s/n;

}printf("Average of \n");float max=avg[0],min=avg[0];float s=0;

for(i=0;i<3;i++)

{printf("Quiz %d=>=%f\n",i+1,avg[i]);s=s+avg[i];

if(avg[i]>max)

    {max=avg[i];

    }

    if(avg[i]<min)

    {min=avg[i];

    }

}

printf("Overall average of all three quizzes=%f\n",s/3);

printf("Highest average = Quiz no.");

for(i=0;i<3;i++)

if(max==avg[i])printf("%d",i+1);

```

```

printf("\b=%f\nLowest average = Quiz no.",max);

for(i=0;i<3;i++)

if(min==avg[i])printf("%d",i+1);

printf("\b=%f\n",min);

for(i=0;i<n;i++)

{int max=b[0][i],min=b[0][i];

    for(j=1;j<3;j++)

    {

        if(b[j][i]>max)

            max=b[j][i];

        if(b[j][i]<min)

            min=b[j][i];

    }printf("Student %d =>\nhighest score=%d Quiz no. ",i+1,max);

    for(j=0;j<3;j++)

    if(b[j][i]==max)printf("%d",j+1);

    printf("\b \nlowest score=%d Quiz no.",min);

    for(j=0;j<3;j++)

    if(b[j][i]==min)printf("%d",j+1);

    printf("\b \n");

}

return 0;

}

```

13. WAP to input 2 matrixes and multiply them.

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

```
int main()
```

```

{int m,n,p,q;

scanf("%d%d%d%d",&m,&n,&p,&q);

int a[m][n],b[p][q],c[m][q];

printf("enter 1st matrix\n");

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

scanf("%d",&a[i][j]);

printf("enter 2nd matrix\n");

for(int i=0;i<p;i++)

for(int j=0;j<q;j++)

scanf("%d",&b[i][j]);

if(n==p)

{

    for(int i=0;i<m;i++)

    {

        for(int j=0;j<q;j++)

        {int s=0;

            for(int r=0;r<n;r++)

            s=s+a[i][r]*b[r][j];

            c[i][j]=s;

        }

    }

printf("A×B=\n");

    for(int i=0;i<m;i++)

    {

```

```

        for(int j=0;j<q;j++)
printf("%d\t",c[i][j]);

printf("\n");

    }

}

else

printf("cant multiply");

    return 0;

}

```

14. WAP to input a matrix and print its upper triangular matrix.

```

#include<stdio.h>

int main()

{

    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];

    if(m==n)

    {

        for(i=0;i<m;i++)

        {

            for(j=0;j<n;j++)

            {

                scanf("%d",&a[i][j]);

            }

        }

    }

}

```

```

    }

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(j>=i)printf("%d\t",a[i][j]);

            else printf("0\t");

        }printf("\n");
    }
}

else printf("Invalid Input");

return 0;

}

```

15. WAP to input a matrix and print its Lower triangular matrix.

```

#include<stdio.h>

int main()
{
    int m,n,i,j;

    scanf("%d%d",&m,&n);

    int a[m][n];

    if(m==n)

    {
        for(i=0;i<m;i++)

        {
            for(j=0;j<n;j++)

```



```

        {
            scanf("%d",&a[i][j]);

        }
    }

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(j<=i)printf("%d\t",a[i][j]);
            else printf("0\t");
        }printf("\n");
    }
}

else printf("Invalid Input");

return 0;

}

```

16. WAP to input a matrix and print it in zig zack form.

Example :- if matrix is

1	2	3	4
5	6	7	8
9	1	2	3
4	5	9	1

Then output matrix will be

1	2	3	4
8	7	6	5
9	1	2	3
1	9	5	4

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

```
int main()
```

```

{
    int m, n, i, j;

    scanf("%d%d", &m, &n);

    int a[m][n];

    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }

    for(i=0; i<m; i++)
    {if(i%2==0)
    {
        for(j=0; j<n; j++){

            printf("%d ", a[i][j]);}

        }
    else
    {
        for(j=n-1; j>=0; j--)
        {

            printf("%d ", a[i][j]);}

        }
    }
}

```

```

        printf("\n");
    }

    return 0;
}

```

17. Suppose there is a game known as “MATCH THE TABLES”, in which the player picks up two tables(each having 10 rows and 10 columns) and matches them. If out of 100 entries at least 90 corresponding entries match then the tables are said to be identical and the player is declared the winner. Wap in ‘C’ to implement the above game.

```

#include<stdio.h>

int main()
{
    int i,j;

    int a[10][10];int b[10][10];int c=0;

    for(i=0;i<10;i++)
    {
        for(j=0;j<10;j++)
        {
            scanf("%d%d",&a[i][j],&b[i][j]);

            if(a[i][j]==b[i][j])c++;
        }
    }

    if(c>=90)printf("Winner");

    else printf("Loser");

    return 0;
}

```

18. Which of the following initializations of a 2d array are valid?

- (i) `int abc[2][2] = { 1, 2, 3 ,4 }` -VALID (ii) `int abc[][] = { 1, 2, 3 ,4 }` -NOT VALID

(iii) `int abc[][] = {1, 2, 3, 4}` - NOT VALID

(iv) `int abc[2][] = {1, 2, 3, 4}` - NOT VALID

19. An array `X [-15.....10, 15.....40]` requires one byte of storage. If beginning location is 1500 determine the location of `X [5][20]` for data stored as

(i) Column major wise

Handwritten solution for column-major storage:

Q. A. (i) Column Major Wise
B.A. = 1500, W = 1B
 $LBR = -15, UBR = 10, LBC = 15, UBC = 40$
No. of Rows, $R = UBR - LBR + 1$
 $= 10 - (-15) + 1 = 26$
Add. of `X[5][20] = B.A. + W[(I - LBR) + R(J - LBC)]`
 $= 1500 + 1[(5 - (-15)) + 26(20 - 15)]$
 $= 1500 + (20 + 26 \times 5)$
 $= 1650$ Ans

(ii) Row major wise.

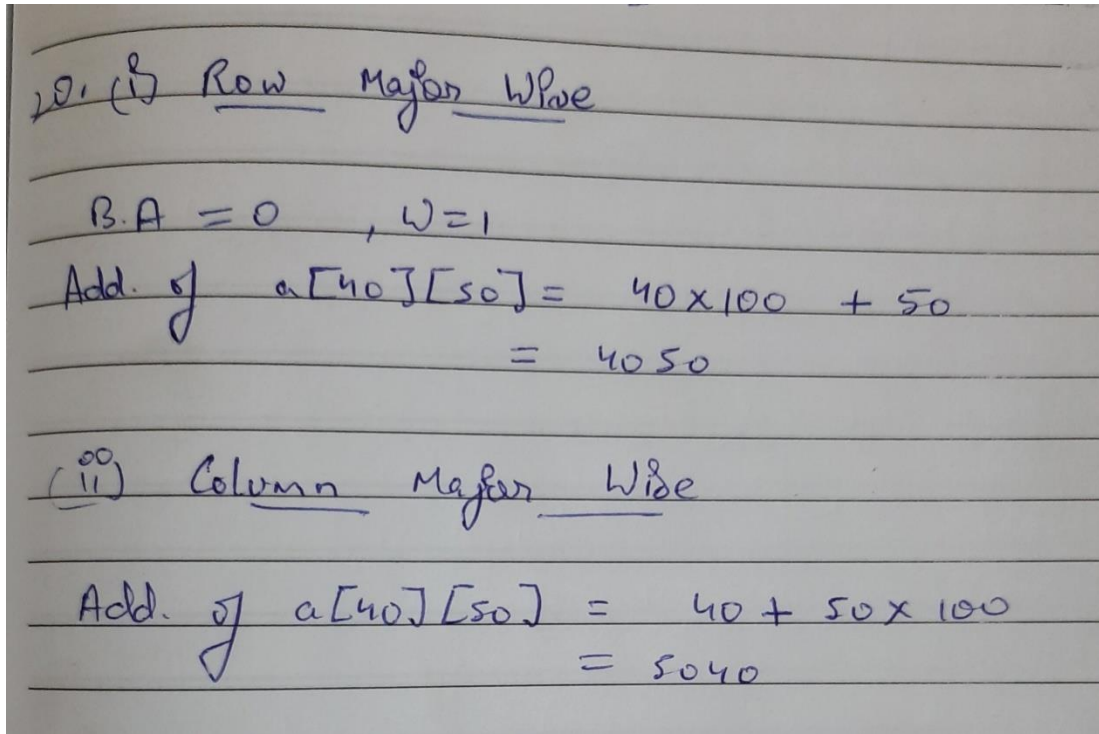
Handwritten solution for row-major storage:

Q. A. (ii) Row Major Wise
DATE/...../.....
No. of columns = $UBC - LBC + 1$
 $= 40 - 15 + 1 = 26$
Add. of `X[5][20] = B.A. + W[(I - LBR) \times 26 + (J - LBC)]`
 $= 1500 + (20 \times 26 + 5)$
 $= 2025$ Ans

20. Consider the following declaration of a 'two-dimensional array in C:

`char a[100][100];`

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0. Find the address of $a[40][50]$ using row and column major.



21. Let A be a square matrix of size $n \times n$. Consider the following program. What is the expected output?

```
C = 100;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
    {
        Temp = A[i][j] + C;
        A[i][j] = A[j][i];
        A[j][i] = Temp - C;
    }
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        printf("%d ", A[i][j]);
```

Ans. Same original 2D array will be printed.

