

Assignment-16

```

1. #include <stdio.h>
int main ()
{
    int a[3][3] = { 12, 7, 5,
                    6, 9, 10,
                    8, 5, 13 };
    int b[3][3] = { 9, 1, 7,
                    6, 5, 4,
                    3, 2, 1 };
    int i, j;
    for (i = 0; i <= 2; i++)
    {
        for (j = 0; j <= 2; j++)
            printf ("%d ", a[i][j] + b[i][j]);
        printf ("\n");
    }
    return 0;
}

```

```

3. #include <stdio.h>
int main ()
{
    int a[3][3] = { 5, 1, 9,
                    8, 10, 11,
                    7, 9, 6 };

```

```
int b[3][3] = { 12, 26, 35,
                 1, 7, 9,
                 8, 5, 11 };
```

```
int i, j;
for (i=0; i<=2; i++)
```

```
{ for (j=0; j<=2; j++)
```

```
    printf("%d ", a[i][j] * b[i][j]);
```

```
}
printf("\n");
```

```
return 0;
```

3. #include <stdio.h>

```
int main()
```

```
{
```

```
int a[3][3] = { 5, 4, 6, 7, 8, 9,
```

```
10, 11, 12 };
```

```
int trans[3][3];
```

```
int i, j;
```

```
printf("Matrix elements : \n");
```

```
for (i=0; i<=2; i++)
```

```
{
```

```
    for (j=0; j<=2; j++)
```

```
{
```

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

```
}
```

```
3 printf("\n");
3 for (i=0; i<=2; i++)
    {
        for (j=0; j<=2; j++)
            trans[i][j] = a[i][j];
        printf("\n");
    }
printf("\n");
printf("Transpose of matrix:\n");
for (i=0; i<=2; i++)
    {
        for (j=0; j<=2; j++)
            printf("%d ", trans[i][j]);
        printf("\n");
    }
return 0;
```

```
4- #include <stdio.h>
int main()
{
    int a[3][3] = { 6, 7, 5,
                    9, 11, 15,
                    1, 2, 7 };
}
```

```
int i, j, sum = 0;
printf("Matrix elements are %m");

```

```
for (i=0; i<=2; i++)

```

{

```
    for (j=0; j<=2; j++)

```

{

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

```

}

```
    printf("\n");

```

}

```
printf("\n");

```

```
printf("Sum of Diagonal elements: %m");

```

```
for (i=0; i<=2; i++)

```

{

```
    for (j=0; j<=2; j++)

```

{

```
        if (i==j) sum += a[i][j];

```

}

```
    sum = sum + a[i][j];

```

}

{

```
printf("%d", sum);

```

5- #include <stdio.h>

```
int main()

```

{

```
    int a[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

```

```
    {1, 2, 3, 4, 5, 6, 7, 8, 9};

```

```
    {1, 2, 3, 4, 5, 6, 7, 8, 9};

```

int i, j, sum = 0;

printf(" Matrix Elements are : \n");

for (i=0; i<=2; i++)

{

 for (j=0; j<=2; j++)

{

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

}

 printf("\n");

}

 printf(" Sum of Left diagonal is : \n");

 for (i=0; i<=2; i++)

{

 for (j=0; j<=2; j++)

{

 if (i+j == 2)

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

}

 printf("%d ", sum);

.

6. #include <stdio.h>

int main()

{

 int a[3][3] = { {7, 6, 10},
 {10, 1, 2},
 {5, 9, 1} };
 int i, j, sum = 0, sum1 = 0;

 printf(" Matrix element are : \n");

```
for(i=0; i<=2; i++)
```

{

```
    for(j=0; j<=2; j++)
```

{

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

{

```
    printf("\n");
```

{

```
printf("\n");
```

```
printf("Sum of rows : \n");
```

```
for(i=0; i<=2; i++)
```

{

```
    for(j=0; j<=2; j++)
```

{

```
        sum = sum + a[i][j];
```

{

```
    printf("\n");
```

```
    printf("Sum of % row is %d ", i, sum);
```

```
sum = 0;
```

{

```
printf("\n");
```

```
printf("\n");
```

```
printf("Sum of Columns : \n");
```

```
for(j=0; j<=2; j++)
```

{

```
    for(i=0; i<=2; i++)
```

{

```
        sum1 = sum1 + a[j][i];
```

{

```
    printf("\n");
```

```
printf("Sum of %d column is %d, %d\n", sum);
sum = 0;
```

3.

```
#include <stdio.h>
int main()
```

{

```
int a[3][3] = { 12, 5, 8,
                 7, 9, 3,
                 5, 3, 1 };
```

int i, j;

printf("Matrix elements are:\n");

for (i=0; i<=2; i++)
 {

```
        for (j=0; j<=2; j++)
            printf("%d ", a[i][j]);
    }
```

printf("\n");

printf("\n");

printf("Lower Triangular Matrix:\n");

for (i=0; i<=2; i++)
 {

```
        for (j=0; j<=2; j++)
            if (j<=i)
                printf("%d ", a[i][j]);
    }
```

if (j<=i)

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

else

printf(" ") ;

3.

printf("\n") ;

3.

#include <stdio.h>

int main()

{

a[3][3] = { 12, 5, 8,

7, 9, 3,

5, 3, 1 } ;

int i, j ;

printf(" Matrix Elements are :\n") ;

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

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

}

printf("\n") ;

3.

printf("\n") ;

printf(" Upper Triangular Matrix are :\n") ;

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

if(j>=i)



Date _____

Page 9

printf("%d", arr[0]) ; which
else

printf(" ") ;

< d. after abutment

printf("\n") ;

: (t[i] - t[i-1]) down bias

: min bias

if ("zoom out offset") then
if & of "bx bx" then
: (dx - dx) down
if & "bx = d" : bx = 0 then
: 0 down

(p + bias - dt bias) down bias

: t bias

: p bias = t

: p bias = dt

: t = dt

< d. after abutment

: (bias t[i] - t[i-1]) up bias

: (bias t[i] - p * t[i]) self bias

: (bias t[i] - 2 * t[i]) from bias

: 0 down bias

: (bias t[i] - 2 * t[i-1]) up bias

: (bias t[i] - 2 * t[i-1]) self bias

pointf("Y, q ", C[1]) ;

g
3 // end of main .

As-16 Q. 9. Sparse Matrix or not

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

```
int main( )
```

S

```
int a[3][3] = { 3, 0, 0,  
                0, 5, 0,  
                6, 0, 0 } ;
```

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

S

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

S

```
    if(a[i][j] != 0)
```

```
        count++ ;
```

3

```
if (count > (9/2))
```

```
    pointf(" Not Sparse Matrix");
```

else

```
    pointf(" Sparse Matrix");
```

g.