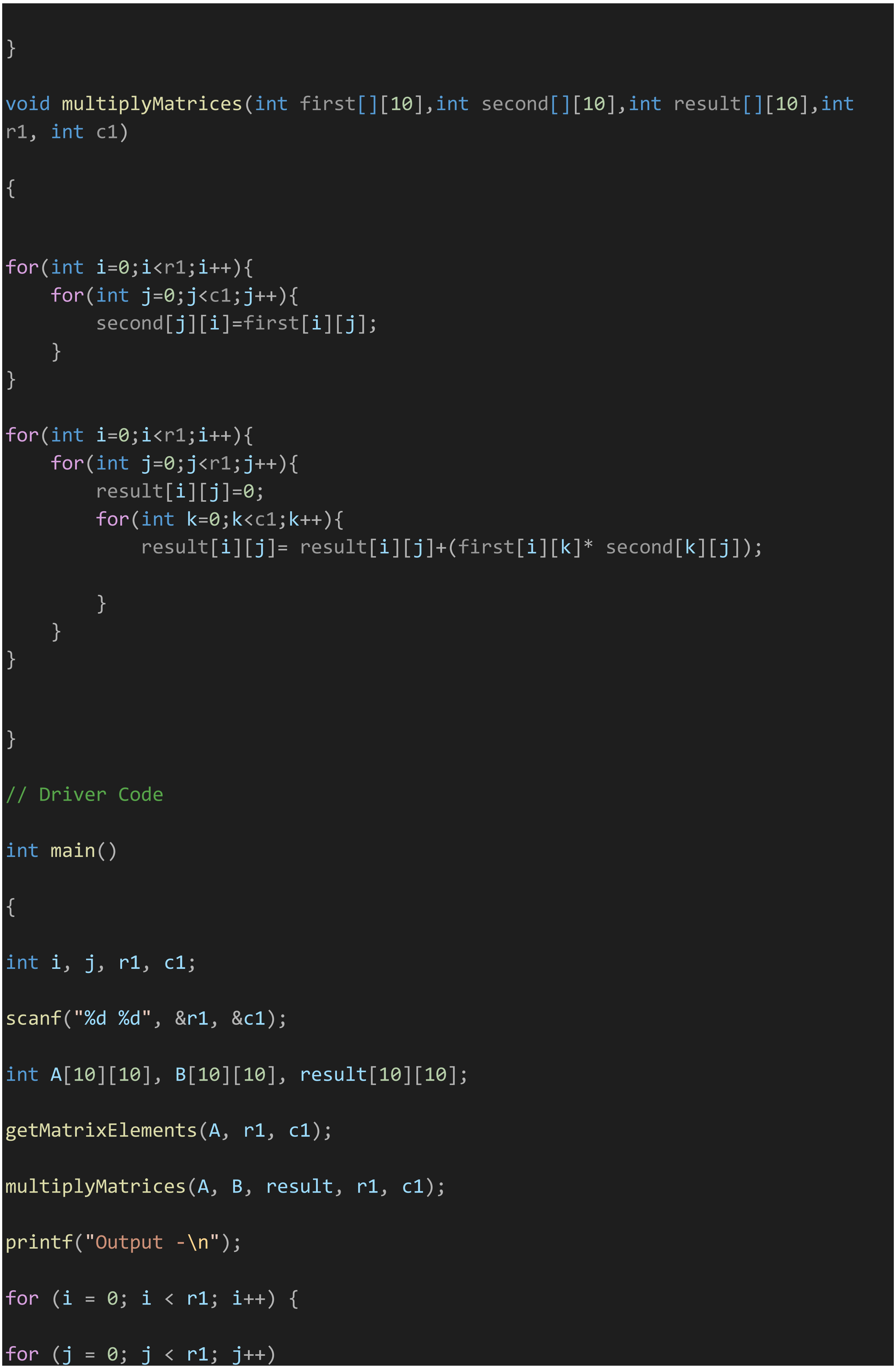
Unit - 1 (Introduction, Asymptotic Notations, Recursive functions)

Ticking Session-1

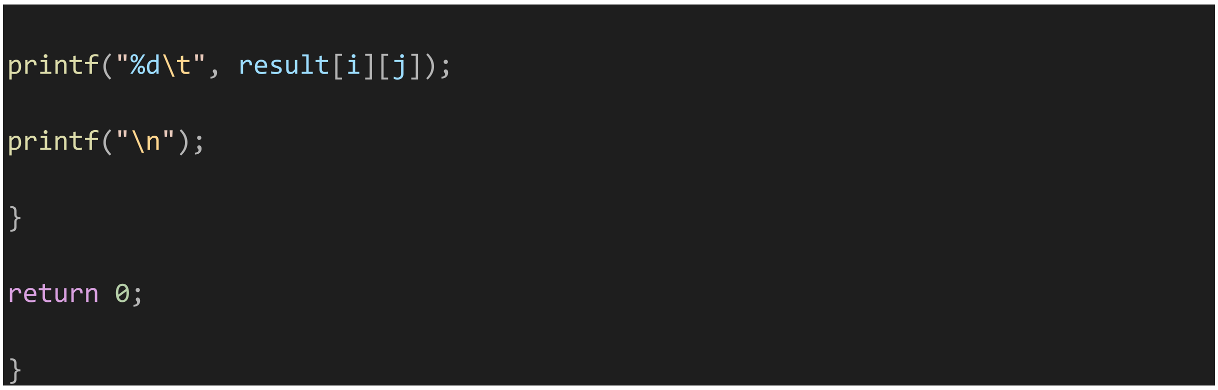
NAGAVENI L G
PES2UG21CS315
4TH SEM - F SEC

1. A matrix M is obtained by multiplying the given matrix with the matrix obtained by flipping the given matrices elements about the principal diagonal. The flipped matrix is found to have the same diagonal elements as the given matrix in the same positions.

```
#include <stdio.h>

void getMatrixElements(int matrix[][10], int row, int column)

{

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

{

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

{

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

}

}
```

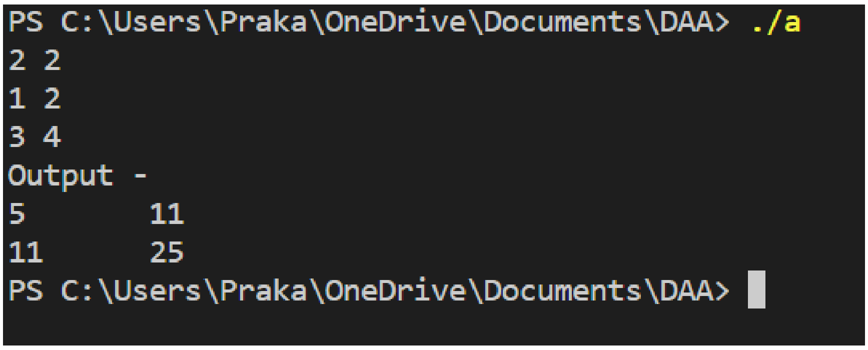


```
}

void multiplyMatrices(int first[][10],int second[][10],int result[][10],int
r1, int c1)

{


for(int i=0;i<r1;i++){
    for(int j=0;j<c1;j++){
        second[j][i]=first[i][j];
    }
}

for(int i=0;i<r1;i++){
    for(int j=0;j<r1;j++){
        result[i][j]=0;
        for(int k=0;k<c1;k++){
            result[i][j]= result[i][j]+(first[i][k]* second[k][j]);

        }
    }
}


}

// Driver Code

int main()

{

int i, j, r1, c1;

scanf("%d %d", &r1, &c1);

int A[10][10], B[10][10], result[10][10];

getMatrixElements(A, r1, c1);

multiplyMatrices(A, B, result, r1, c1);

printf("Output -\n");

for (i = 0; i < r1; i++) {

for (j = 0; j < r1; j++)
```

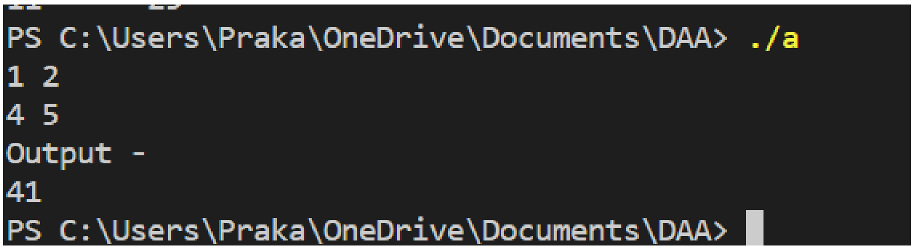


```
printf("%d\t", result[i][j]);

printf("\n");

}

return 0;

}
```

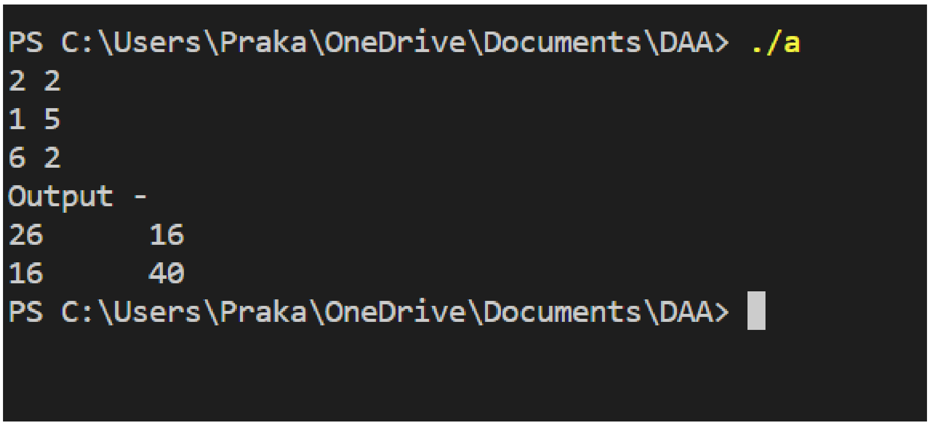
TESTCASES



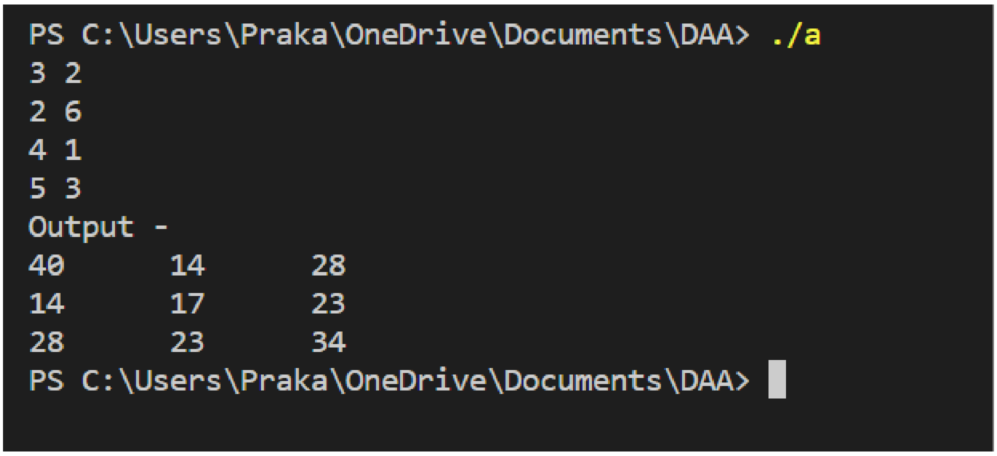
```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
2 2
1 2
3 4
Output -
5       11
11      25
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```



```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
1 2
4 5
Output -
41
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```

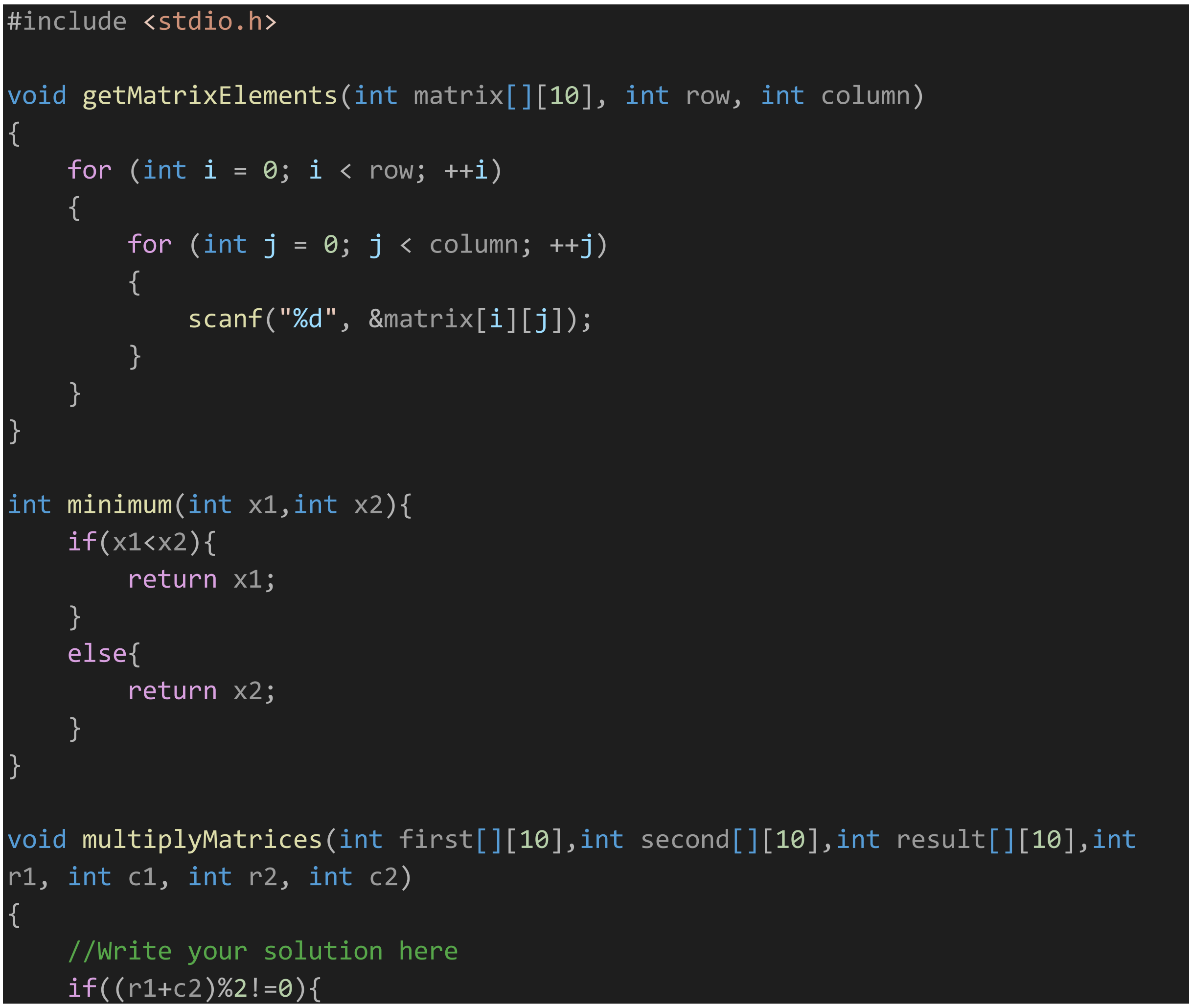


```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
2 2
1 5
6 2
Output -
26      16
16      40
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```

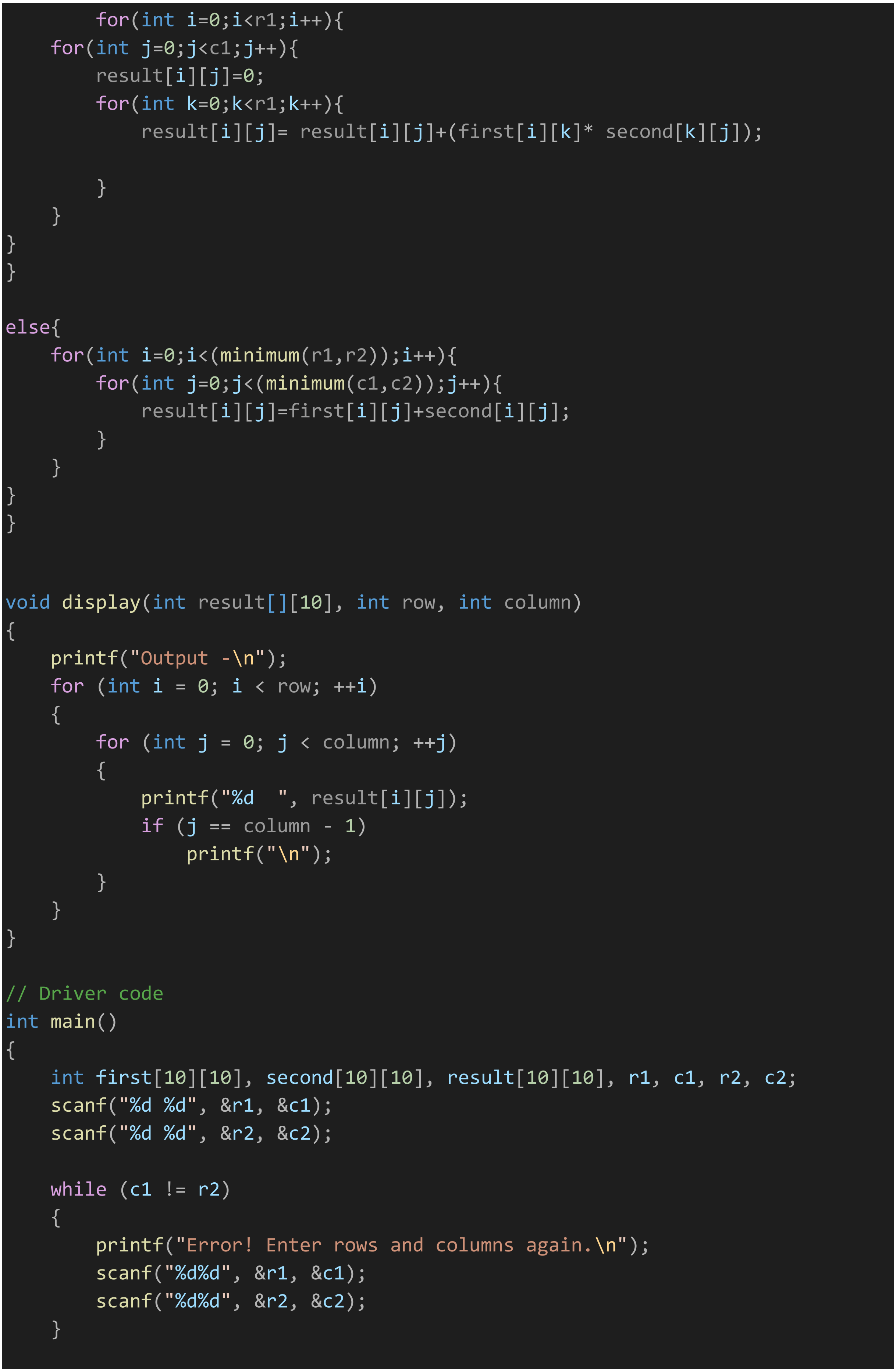


```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
3 2
2 6
4 1
5 3
Output -
40      14      28
14      17      23
28      23      34
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```

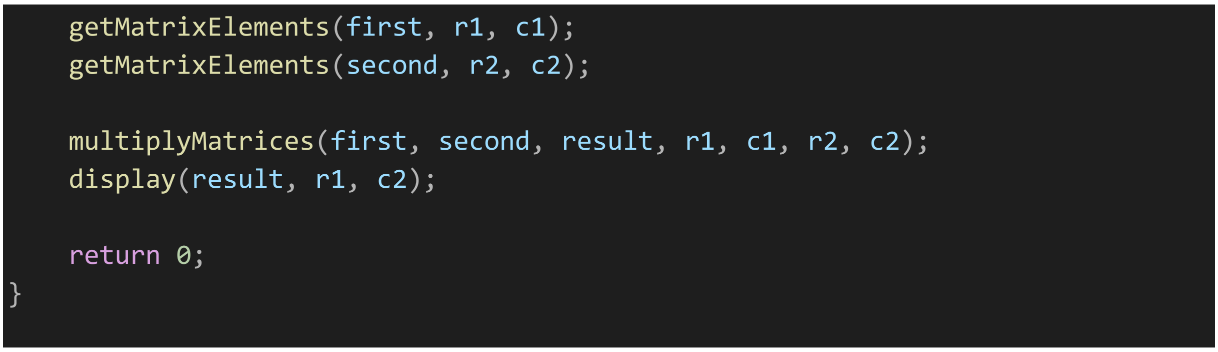
2. If the sum of the number of rows of matrix1 and number of columns of matrix2 is not divisible by the smallest prime number, then perform matrix multiplication else perform matrix addition.



```c
#include <stdio.h>

void getMatrixElements(int matrix[][10], int row, int column)
{
    for (int i = 0; i < row; ++i)
    {
        for (int j = 0; j < column; ++j)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}

int minimum(int x1,int x2){
    if(x1<x2){
        return x1;
    }
    else{
        return x2;
    }
}

void multiplyMatrices(int first[][10],int second[][10],int result[][10],int
r1, int c1, int r2, int c2)
{
    //Write your solution here
    if((r1+c2)%2!=0){
```

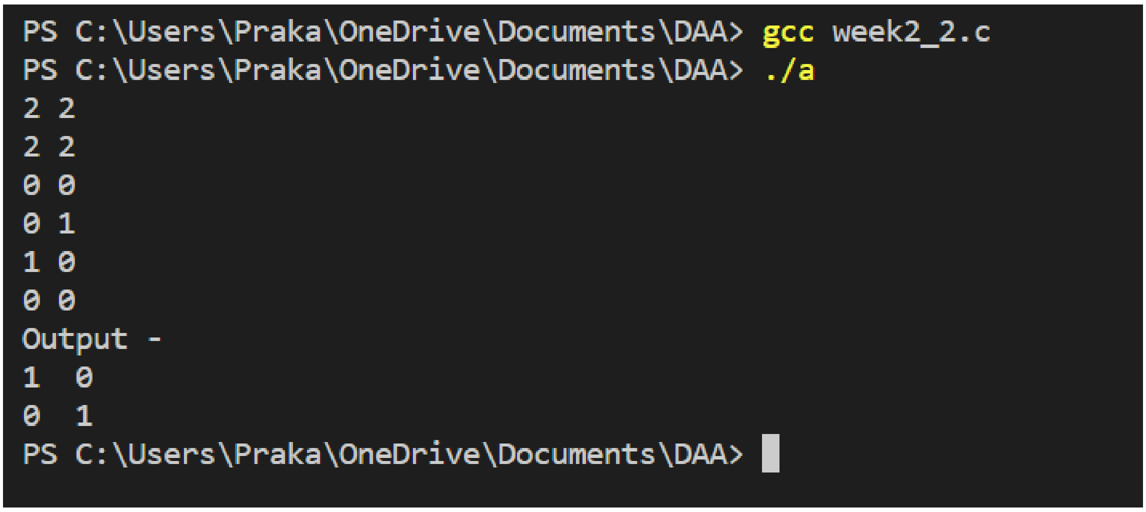


```
        for(int i=0;i<r1;i++){
    for(int j=0;j<c1;j++){
        result[i][j]=0;
        for(int k=0;k<r1;k++){
            result[i][j]= result[i][j]+(first[i][k]* second[k][j]);

        }
    }
}
}

else{
    for(int i=0;i<(minimum(r1,r2));i++){
        for(int j=0;j<(minimum(c1,c2));j++){
            result[i][j]=first[i][j]+second[i][j];
        }
    }
}
}


void display(int result[][10], int row, int column)
{
    printf("Output -\n");
    for (int i = 0; i < row; ++i)
    {
        for (int j = 0; j < column; ++j)
        {
            printf("%d  ", result[i][j]);
            if (j == column - 1)
                printf("\n");
        }
    }
}

// Driver code
int main()
{
    int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;
    scanf("%d %d", &r1, &c1);
    scanf("%d %d", &r2, &c2);

    while (c1 != r2)
    {
        printf("Error! Enter rows and columns again.\n");
        scanf("%d%d", &r1, &c1);
        scanf("%d%d", &r2, &c2);
    }
```

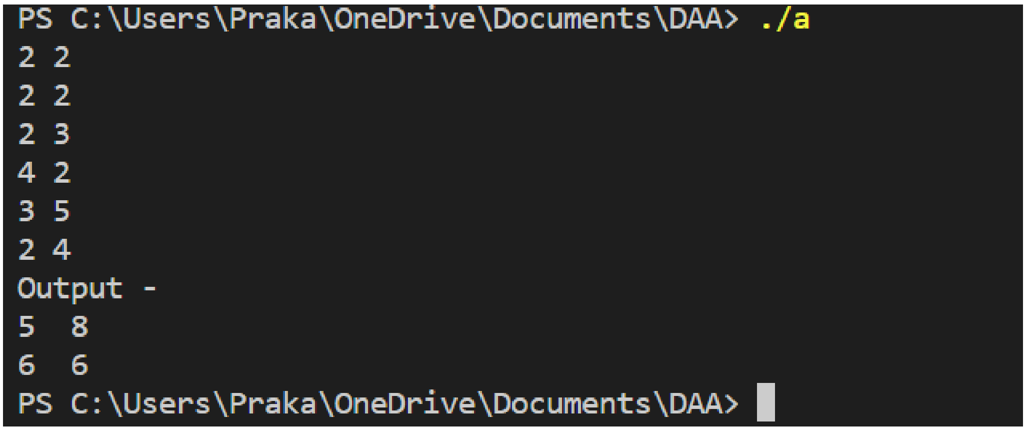


```
    getMatrixElements(first, r1, c1);
    getMatrixElements(second, r2, c2);

    multiplyMatrices(first, second, result, r1, c1, r2, c2);
    display(result, r1, c2);

    return 0;
}
```

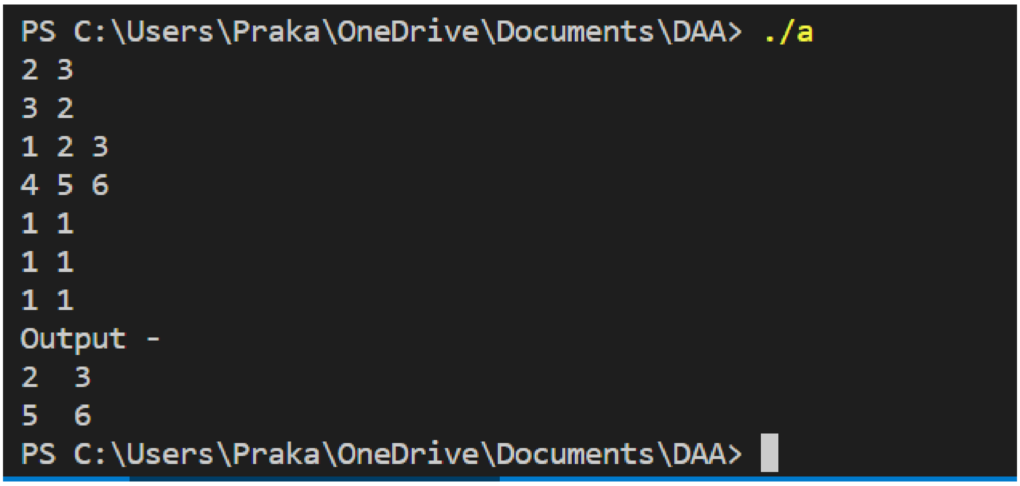
TESTCASES



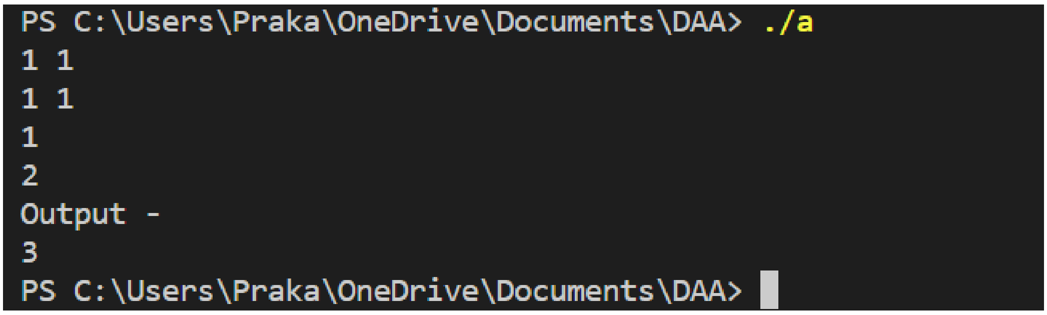
```
PS C:\Users\Praka\OneDrive\Documents\DAA> gcc week2_2.c
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
2 2
2 2
0 0
0 1
1 0
0 0
Output -
1  0
0  1
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```



```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
2 2
2 2
2 3
4 2
3 5
2 4
Output -
5  8
6  6
PS C:\Users\Praka\OneDrive\Documents\DAA> 
```



```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
2 3
3 2
1 2 3
4 5 6
1 1
1 1
1 1
Output -
2  3
5  6
PS C:\Users\Praka\OneDrive\Documents\DAA>
```



```
PS C:\Users\Praka\OneDrive\Documents\DAA> ./a
1 1
1 1
1
2
Output -
3
PS C:\Users\Praka\OneDrive\Documents\DAA>
```

THANK YOU 😊