

Assignment 4.2

Date:22/09/2025

1. Calculate average marks of 10 subjects

```
#include<stdio.h>

void main(){

float marks[10],sum=0,average;

int i;

printf("Enter Marks for 10 subjects\n");

for(i=0;i<10;i++){

    printf("Subject %d=",i+1);

    scanf("%f",&marks[i]);

    sum=sum+marks[i];

}

average=sum/10;

printf("Average of marks is = %f ",average);

}
```

Output :

```
C:\Users\Student\Desktop\codes\avg_array.exe
Enter Marks for 10 subjects
Subject 1=54.65
Subject 2=45
Subject 3=87
Subject 4=98
Subject 5=67
Subject 6=56
Subject 7=45.6
Subject 8=34.8
Subject 9=98.7
Subject 10=65
Average of marks is = 65.175003
Process returned 32 (0x20)    execution time : 41.585 s
Press any key to continue.
```

2. Matrix addition and multiplication

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

```
void main(){
```

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

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

```
    int sum[3][3];
```

```
    int i, j;
```

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

```
        for(j=0;j<3;j++){  
            sum[i][j] = a[i][j] + b[i][j];  
        }  
  
    }
```

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

```
for(i=0;i<3;i++){  
    for(j=0;j<3;j++){  
        printf("%d\t",sum[i][j]);  
    }  
    printf("\n");  
}
```

```
for(i=0;i<3;i++){  
    for(j=0;j<3;j++){  
        sum[i][j] = a[i][j] * b[i][j];  
    }  
  
}
```

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

```
for(i=0;i<3;i++){  
    for(j=0;j<3;j++){  
        printf("%d\t",sum[i][j]);  
    }  
    printf("\n");  
}  
}
```

Output:

```
Matrix elements after adding :  
6      4      2  
8      10     12  
9      12     16  
Matrix elements after multiplying :  
9      4      1  
16     25     36  
20     36     64  
  
Process returned 10 (0xA)   execution time : 0.101 s  
Press any key to continue.
```

B) Matrix multiplication

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

```
void main() {
```

```
    int m1[3][2] = {{1, 2}, {3, 4}, {5, 6}};
```

```
    int m2[2][3] = {{7, 8, 9}, {10, 11, 12}};
```

```
    int r1 = 3, c1 = 2;
```

```
    int r2 = 2, c2 = 3;
```

```
    int result[r1][c2];
```

```
    if (c1 != r2) {
```

```
        printf("Matrix multiplication is not possible.\n");
```

```
        return;
```

```
    }
```

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

```
        for (int j = 0; j < c2; j++) {
```

```
            result[i][j] = 0;
```

```
    }  
}
```

```
for (int i = 0; i < r1; i++) {  
    for (int j = 0; j < c2; j++) {  
        for (int k = 0; k < c1; k++) {  
            result[i][j] += m1[i][k] * m2[k][j];  
        }  
    }  
}
```

```
printf("Matrix 1:\n");  
for (int i = 0; i < r1; i++) {  
    for (int j = 0; j < c1; j++) {  
        printf("%d\t", m1[i][j]);  
    }  
    printf("\n");  
}
```

```
printf("\nMatrix 2:\n");  
for (int i = 0; i < r2; i++) {  
    for (int j = 0; j < c2; j++) {
```

```

        printf("%d\t", m2[i][j]);

    }

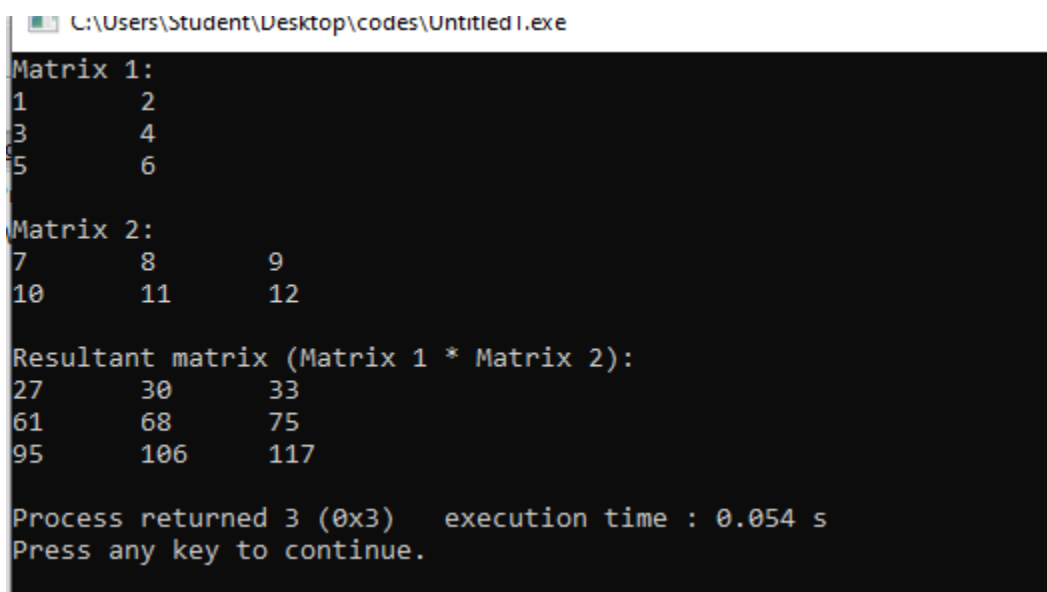
    printf("\n");
}

printf("\nResultant matrix (Matrix 1 * Matrix 2):\n");

for (int i = 0; i < r1; i++) {
    for (int j = 0; j < c2; j++) {
        printf("%d\t", result[i][j]);
    }

    printf("\n");
}
}

```



```

C:\Users\Student\Desktop\codes\Untitled1.exe
Matrix 1:
1      2
3      4
5      6

Matrix 2:
7      8      9
10     11     12

Resultant matrix (Matrix 1 * Matrix 2):
27     30     33
61     68     75
95     106    117

Process returned 3 (0x3)   execution time : 0.054 s
Press any key to continue.

```

3. Transpose of a matrix

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

```
void main() {
```

```
    int matrix[4][4] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16}};
```

```
    int transpose[4][4], i, j;
```

```
    printf("Elements of matrix before transpose calculation...\n");
```

```
    for (i = 0; i < 4; i++) {
```

```
        for (j = 0; j < 4; j++) {
```

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

```
        }
```

```
    }
```

```
    printf("\nElements of matrix after transpose calculation...\n");
```

```
    for (i = 0; i < 4; i++) {
```

```
        for (j = 0; j < 4; j++) {
```

```
            transpose[i][j] = matrix[j][i];
```

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

```
        }
```



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

Output:

```
Elements of matrix before transpose calculation...  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
Elements of matrix after transpose calculation...  
1      5      9      13  
2      6      10     14  
3      7      11     15  
4      8      12     16  
  
Process returned 10 (0xA)    execution time : 0.092 s  
Press any key to continue.
```