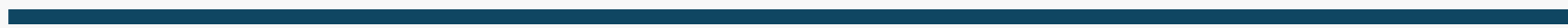


Math Series

C Programming

UG Sem-3 Major (Kalyani University)

Day - 08



Transpose Matrix

Transpose of a Matrix

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3} \qquad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}_{3 \times 2}$$

Matrix Addition

```
// adding two matrices
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        sum[i][j] = a[i][j] + b[i][j];
    }
```

Transpose of Rectangular Matrix

```
int m = 2, n = 3;

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

int t[3][2];

// Find transpose
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        t[j][i] = a[i][j];
    }
}
```

```
// Display transpose
printf("Transpose of the matrix:\n");
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < m; j++)
    {
        printf("%d ", t[i][j]);
    }
    printf("\n");
}
```

Transpose of Square Matrix

	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

```
// Transpose in-place
for (int i = 0; i < n; i++) {
    for (int j = i; j < n; j++) {
        int temp = a[i][j];
        a[i][j] = a[j][i];
        a[j][i] = temp;
    }
}
```

Transpose of Square Matrix

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

// Transpose in-place
for (int i = 0; i < n; i++) {
    for (int j = i; j < n; j++) {
        int temp = a[i][j];
        a[i][j] = a[j][i];
        a[j][i] = temp;
    }
}
```

```
// Display transpose
printf("Transpose of the matrix:\n");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        printf("%d ", a[i][j]);
    }
    printf("\n");
}
```

[Home](#)
[About Us](#)
[Our Services](#)
[Our Clients](#)
[Our Partners](#)
[Our Projects](#)
[Our News](#)
[Our Contact](#)

1	1
2	2

1	1
2	2

3	3
6	6

 $(m \times p)$

1	1
2	2
3	3

1	1	1
2	2	2

3	3	3
6	6	6
9	9	9

(3 x 3)

Sum of prime numbers in a range

```
printf("Enter a positive integer: ");
scanf("%d", &num);

if (num < 2) {
    printf("No prime numbers exist below %d.\n", num);
    return 0;
}

printf("Prime numbers between 1 and %d:\n", num);

for (int i = 2; i <= num; i++) {
    if (isPrime(i)) {
        printf("%d ", i);
        sum += i;
        count++;
    }
}

printf("\n\nTotal prime numbers found: %d\n", count);
printf("Sum of all prime numbers: %d\n", sum);
```

```
// function to check if a number is prime
int isPrime(int n) {
    if (n <= 1) {
        return 0; // not prime
    }
    if (n == 2) {
        return 1; // 2 is prime
    }
    if (n % 2 == 0) {
        return 0; // even numbers (except 2) are not prime
    }
    for (int i = 3; i <= sqrt(n); i += 2) {
        if (n % i == 0) {
            return 0; // found a divisor, not prime
        }
    }
    return 1; // prime
}
```