```c
// Program to compute the factorial of a number using recursion.
#include <stdio.h>

// Function to compute the factorial of a number using recursion
int factorial(int);

int main()
{
    int n;

    // Input the number
    printf("Enter a number: ");
    scanf("%d", &n);

    // Calculate the factorial
    int fact = factorial(n);

    // Display the result
    printf("The factorial of %d is %d.\n", n, fact);

    return 0;
}

int factorial(int num)
{
    // Base case: factorial of 0 is 1
    if (num == 0)
        return 1;
    else
        return num * factorial(num - 1);
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$ cd "/home/manish/vs-code/bca-programming-repo/C/recursion_
ay_and_string_to_function/"question1
Enter a number: 7
The factorial of 7 is 5040.
```

```c
// Program to generate Fibonacci series up to n terms using recursive function.
#include <stdio.h>

// Recursive function to generate Fibonacci series
int fibo(int n)
{
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
    else
        return fibo(n - 1) + fibo(n - 2);
}

int main()
{
    int n;

    // Input the number of terms
    printf("Enter n: ");
    scanf("%d", &n);

    // Display Fibonacci numbers up to n terms
    printf("\nFibonacci numbers up to %d terms:\n", n);
    for (int i = 0; i < n; i++)
    {
        printf("%d ", fibo(i));
    }

    return 0;
}
```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$ ./question2
Enter n: 10

Fibonacci numbers up to 10 terms:
0 1 1 2 3 5 8 13 21 34 %
```

```c
// Program to add the first n-natural numbers using recursive function.
#include <stdio.h>

// Recursive function to calculate the sum of the first n natural numbers
int sumOfNatural(int n)
{
    if (n == 0)
        return 0;
    else
        return n + sumOfNatural(n - 1);
}

int main()
{
    int n;

    // Input the number
    printf("Enter a number: ");
    scanf("%d", &n);

    // Calculate and display the sum of the first n natural numbers
    printf("The sum of the first %d natural numbers is %d.\n", n, sumOfNatural(n));

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$ ./question3
Enter a number: 25
The sum of the first 25 natural numbers is 325.

manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$
```

```c
/* Program to read 10 numbers in an array and finds their sum and display
using the function.*/
#include <stdio.h>

// Function to display elements in an array
void output(int arr[])
{
    printf("The elements in the array are:\n");
    for (int i = 0; i < 10; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

// Function to calculate the sum of elements in an array
int sum(int arr[])
{
    int s = 0;
    for (int i = 0; i < 10; i++)
    {
        s += arr[i];
    }
    return s;
}

int main()
{
    int arr[10];

    // Input 10 elements in the array
    printf("Enter 10 elements: ");
    for (int i = 0; i < 10; i++)
    {
        scanf("%d", &arr[i]);
    }

    // Display the elements in the array
    output(arr);

    // Display the sum of the array elements
    printf("Sum of array elements is %d\n", sum(arr));

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$ ./question4
Enter 10 elements: 23 45 67 87 11 12 13 14 15 90
The elements in the array are:
23 45 67 87 11 12 13 14 15 90
Sum of array elements is 377
```

```c
// Program to find the minimum value in an array by passing array to function.
#include <stdio.h>
#define N 10

// Function to find the minimum value in an array
int findMinimum(int arr[])
{
    int minValue = arr[0];
    for (int i = 1; i < N; i++)
    {
        if (minValue > arr[i])
        {
            minValue = arr[i];
        }
    }
    return minValue;
}

int main()
{
    int arr[N];

    // Input 10 numbers into the array
    printf("Enter 10 numbers:\n");
    for (int i = 0; i < N; i++)
    {
        scanf("%d", &arr[i]);
    }

    // Display the minimum value in the array
    printf("Minimum value is %d.\n", findMinimum(arr));

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
$ ./question5
Enter 10 numbers:
22 32 12 4 111 65 35 99 9 10
Minimum value is 4.

manish@fedora: ~/vs-code/bca-programming-repo/C/recursion_pa
```

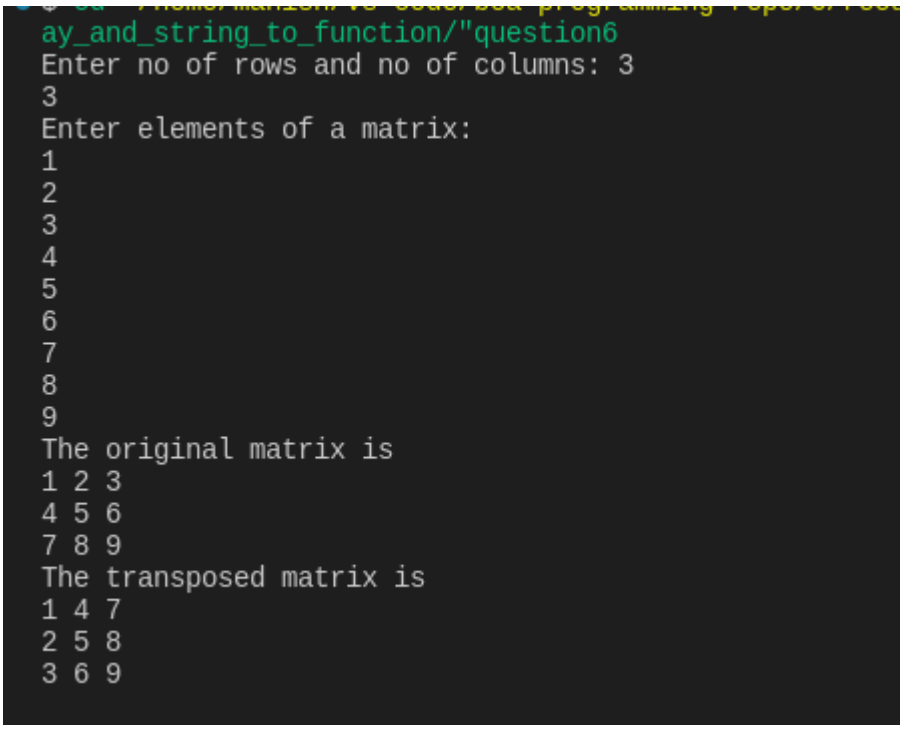
```c
// Program to find the transpose of the matrix using function.
#include <stdio.h>

void display(int[][3], int, int);
void transpose(int[][3], int[][3], int, int);

int main()
{
    int r, c;
    printf("Enter no of rows and no of columns: ");
    scanf("%d%d", &r, &c);
    int matrix[r][c], t[c][r];
    printf("Enter elements of a matrix:\n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("The original matrix is\n");
    display(matrix, r, c);
    transpose(matrix, t, r, c);
    printf("The transposed matrix is\n");
    display(t, c, r);
    return 0;
}

void display(int matrix[][3], int r, int c)
{
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

void transpose(int matrix[][3], int t[][3], int r, int c)
{
    for (int i = 0; i < c; i++)
    {
        for (int j = 0; j < r; j++)
        {
            t[i][j] = matrix[j][i];
        }
    }
}
```


ay_and_string_to_function/"question6
Enter no of rows and no of columns: 3
3
Enter elements of a matrix:
1
2
3
4
5
6
7
8
9
The original matrix is
1 2 3
4 5 6
7 8 9
The transposed matrix is
1 4 7
2 5 8
3 6 9