Name: Pranesh Chowdhury

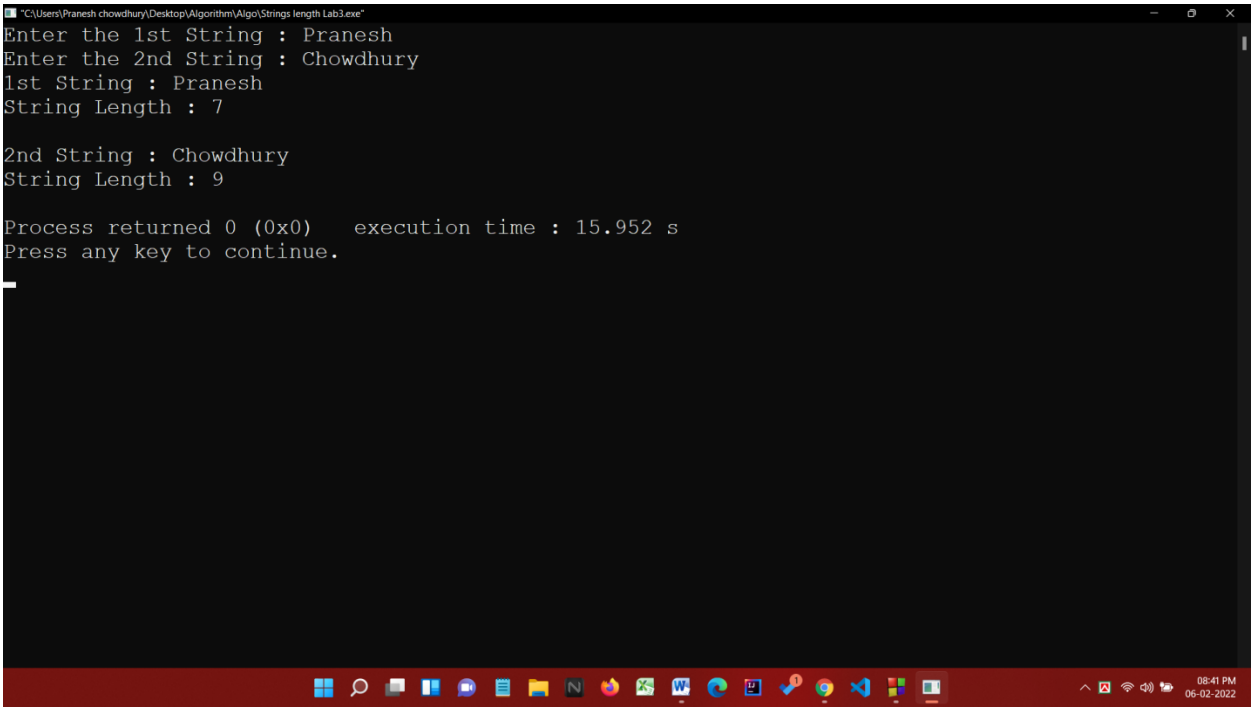
ID: 202003112

Write a C program that takes two strings as input and shows their length as output.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[1000], b[1000];

    printf("Enter the 1st String : ");
    scanf("%s", &a);
    printf("Enter the 2nd String : ");
    scanf("%s", &b);

    int len1 = strlen(a);
    int len2 = strlen(b);

    printf("1st String : %s\n", a);
    printf("String Length : %d\n\n", len1);

    printf("2nd String : %s\n", b);
    printf("String Length : %d\n", len2);

    return 0;
}
```

OUTPUT:



Time Complexity: O(1)

Space Complexity: O(n)

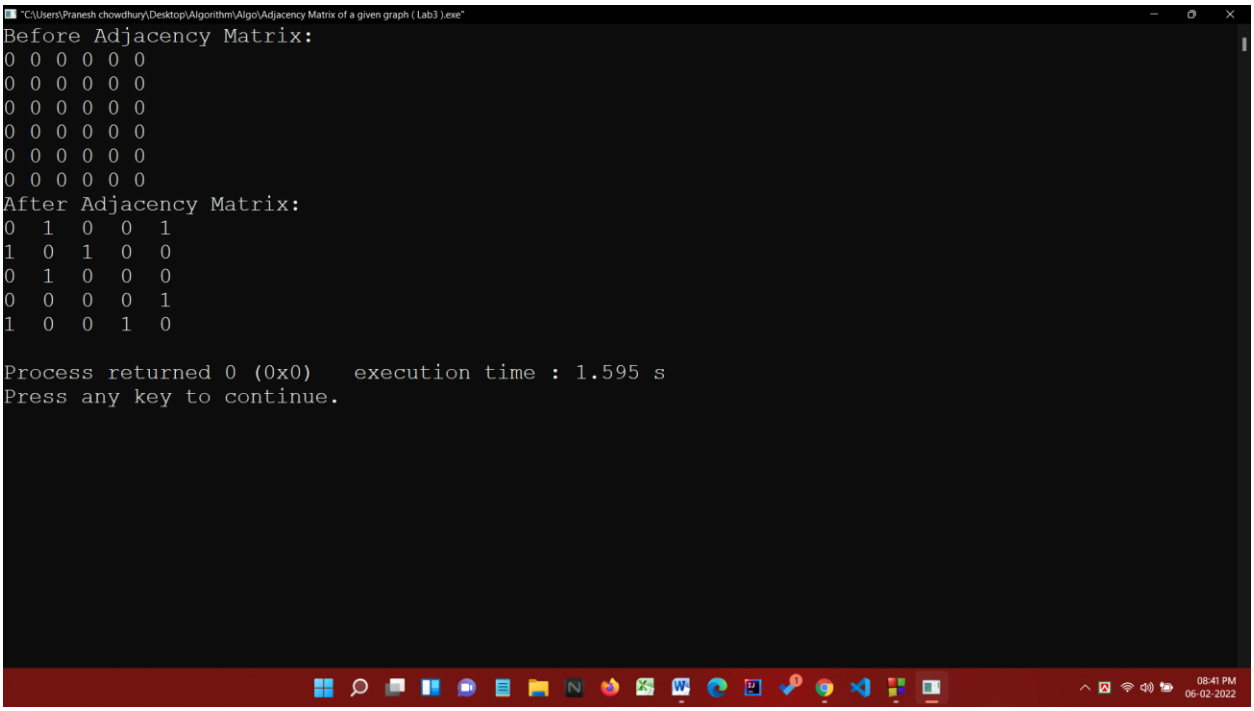
Write a C program to form Adjacency Matrix of a given graph.

PROGRAM:

```
#include<stdio.h>

int main()

{

    int n = 5, m = 4;

    int ara[10][10] = { {1, 2}, {2, 3}, {4, 5}, {1, 5} };

    int adj[10][10];
```

```
    printf("Before Adjacency Matrix: \n");
    for (int i = 0; i < n+1; i++){
        for (int j = 0; j < n+1; j++){
            adj[i][j] = 0;
            printf("%d ", adj[i][j]);
        }
        printf("\n");
    }

    int x, y;
    for (int i = 0; i < n; i++){
            x = ara [i][0];
            y = ara [i][1];
            adj[x][y] = 1;
            adj[y][x] = 1;
    }
    printf("After Adjacency Matrix: \n");
    for (int i = 1; i <= n; i++){
        for (int j = 1; j<= n; j++){
            printf("%d  ", adj[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:



Time Complexity: O(n^2)

Space Complexity: O(n^2)