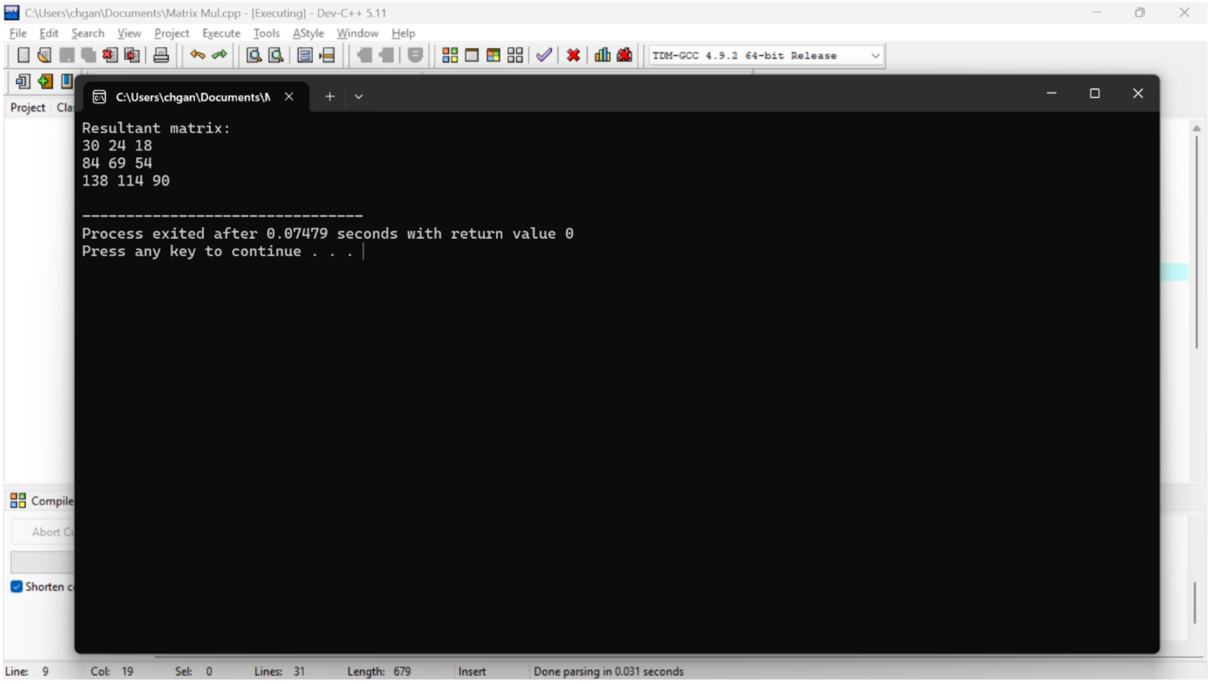
8. Program to perform Matrix multiplication.

PROGRAM:

```
#include <stdio.h>
int main() {
    int a[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int b[3][3] = {
        {9, 8, 7},
        {6, 5, 4},
        {3, 2, 1}
    };
    int result[3][3];
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            result[i][j] = 0;
            for (int k = 0; k < 3; k++) {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    printf("Resultant matrix:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", result[i][j]);
        }
        printf("\n");
```

```
  } }
```

OUTPUT:



C:\Users\chgan\Documents\Matrix Mul.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
Project Cla
C:\Users\chgan\Documents\M
Compile
Abort C
Shorten c
Line: 9 Col: 19 Sel: 0 Lines: 31 Length: 679 Insert Done parsing in 0.031 seconds

```
Resultant matrix:
30 24 18
84 69 54
138 114 90

--------------------------------
Process exited after 0.07479 seconds with return value 0
Press any key to continue . . .
```

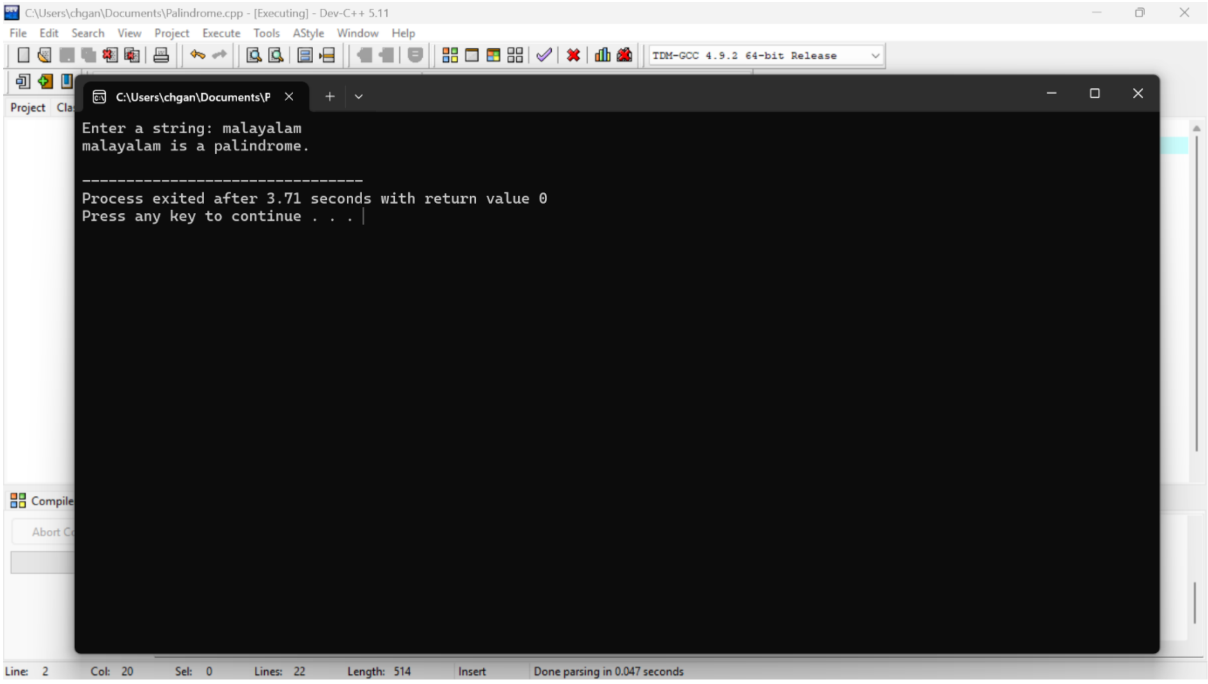
9. Program to check whether a string is Palindrome or not.

PROGRAM:

```
#include <stdio.h>
#include <string.h>
int main() {
   char str[100], rev[100];
   int len, isPalindrome = 1;
   printf("Enter a string: ");
   scanf("%s", str);
   len = strlen(str);
   for (int i = 0; i < len / 2; i++) {
      if (str[i] != str[len - i - 1]) {
         isPalindrome = 0;
         break;
      }
```

```
    }
    if (isPalindrome) {
        printf("%s is a palindrome.\n", str);
    } else {
        printf("%s is not a palindrome.\n", str);
    }
    return 0;
}
```

OUTPUT:


C:\Users\chgan\Documents\Palindrome.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
Project Cla
C:\Users\chgan\Documents\P
Compile
Abort C

```
Enter a string: malayalam
malayalam is a palindrome.

--------------------------------
Process exited after 3.71 seconds with return value 0
Press any key to continue . . .
```

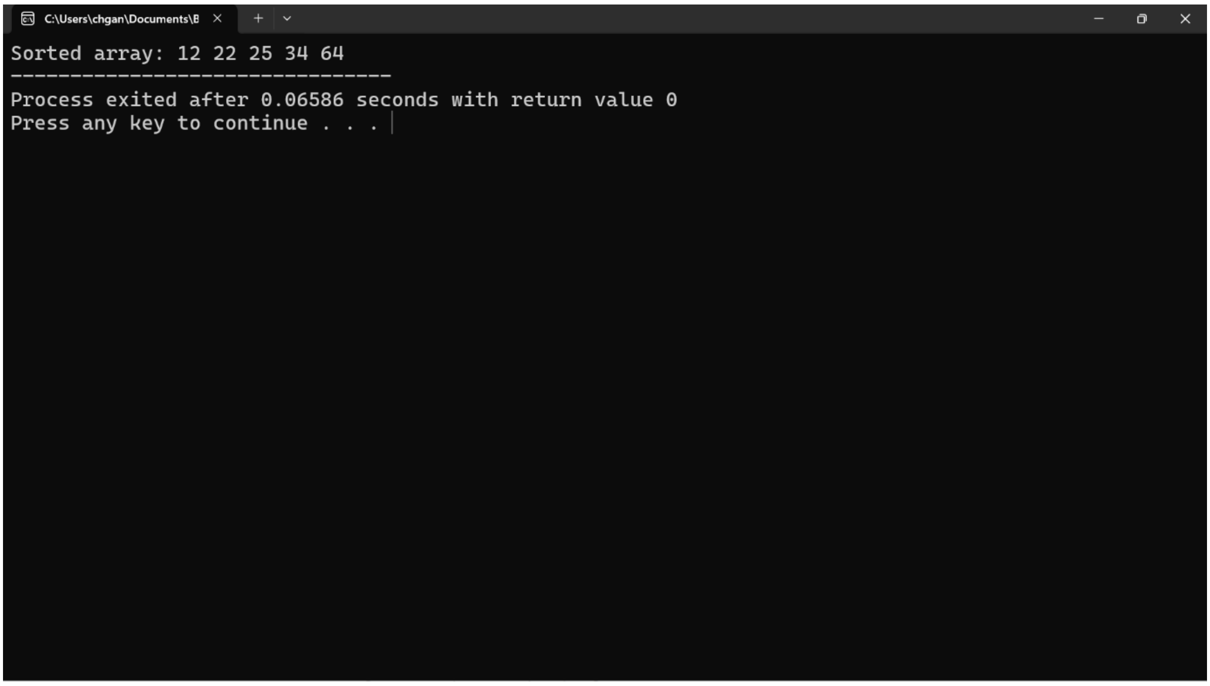
Line: 2 Col: 20 Sel: 0 Lines: 22 Length: 514 Insert Done parsing in 0.047 seconds

10. Program to perform Bubble Sort.

PROGRAM:

```
#include <stdio.h>

int main() {
    int arr[5] = {64, 34, 25, 12, 22};
    int n = 5;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

OUTPUT:



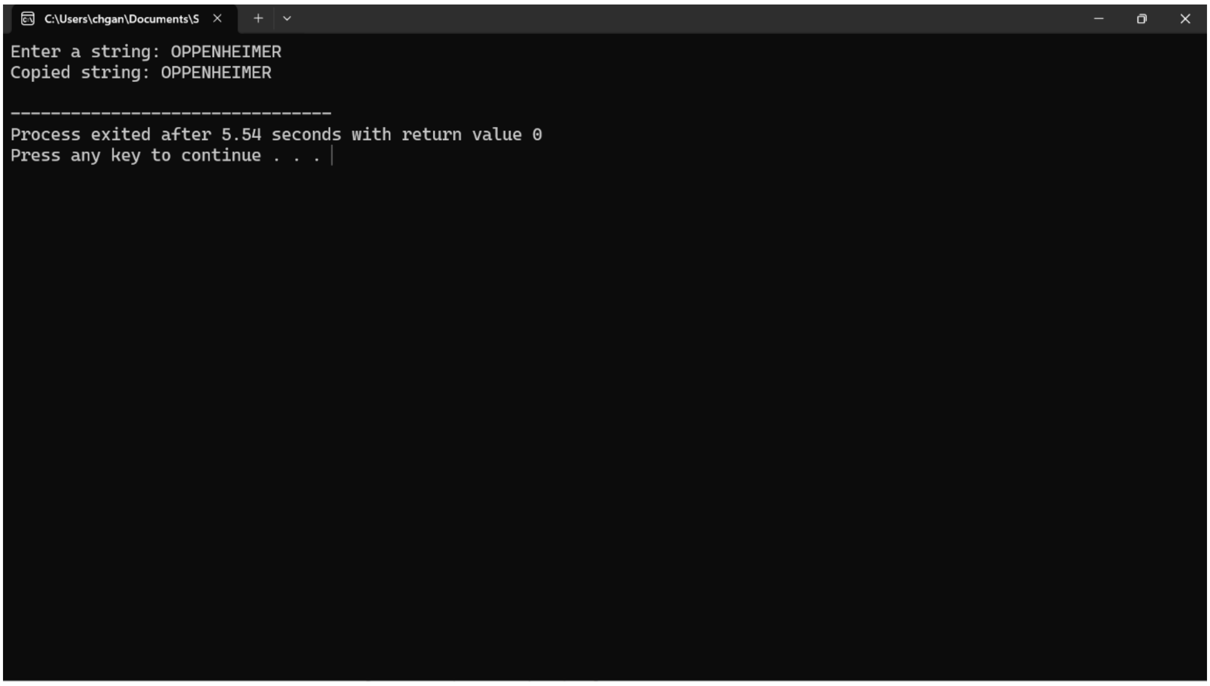
```
Sorted array: 12 22 25 34 64
--------------------------------
Process exited after 0.06586 seconds with return value 0
Press any key to continue . . .
```

11. Program to copy one string to another.

PROGRAM:

```
#include <stdio.h>
int main() {
    char str1[100], str2[100];
    int i;
    printf("Enter a string: ");
    scanf("%s", str1);
    for (i = 0; str1[i] != '\0'; i++) {
        str2[i] = str1[i];
    }
    str2[i] = '\0';
    printf("Copied string: %s\n", str2);
    return 0;
}
```

OUTPUT:



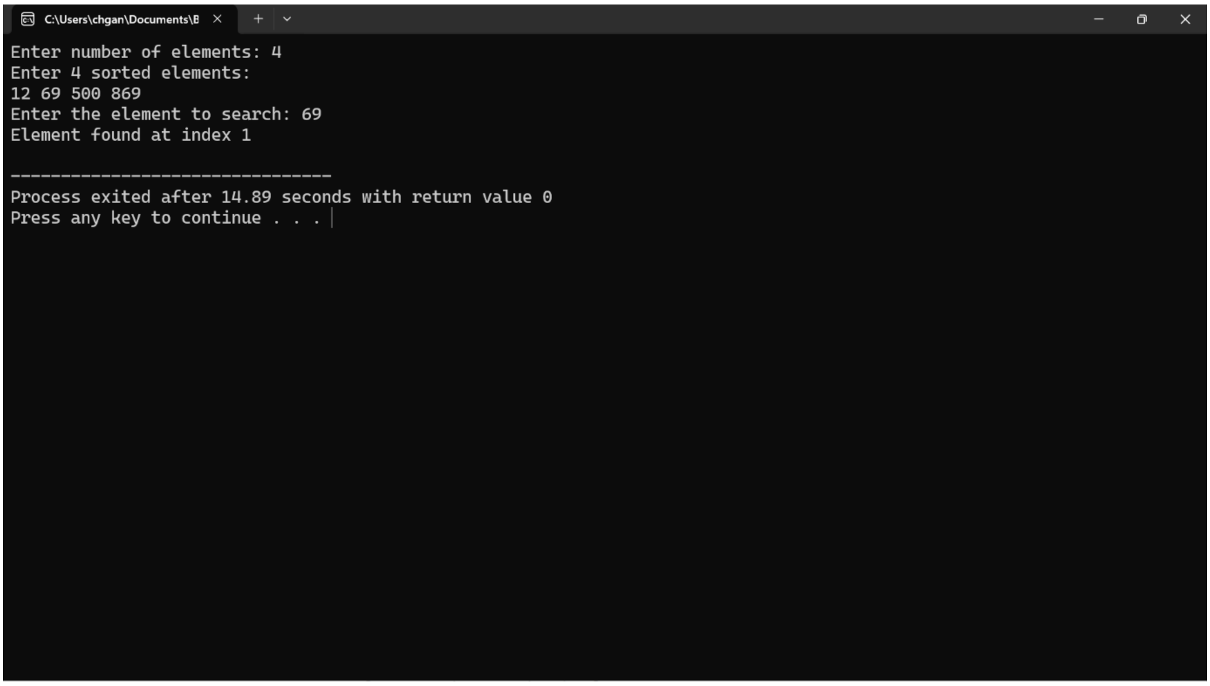
```
Enter a string: OPPENHEIMER
Copied string: OPPENHEIMER

--------------------------------
Process exited after 5.54 seconds with return value 0
Press any key to continue . . .
```

12. Program to perform Binary search in an array of elements.

PROGRAM:

```
#include <stdio.h>
int binarySearch(int arr[], int n, int target) {
    int left = 0, right = n - 1;
    while (left <= right) {
        int mid = (left + right) / 2;
        if (arr[mid] == target) {
            return mid;
        } else if (arr[mid] < target) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    return -1;
}
int main() {
    int arr[100], n, target;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d sorted elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &target);
    int result = binarySearch(arr, n, target);
    if (result != -1) {
        printf("Element found at index %d\n", result);
    } else {
        printf("Element not found\n");
    }
    return 0;
}
```

OUTPUT:



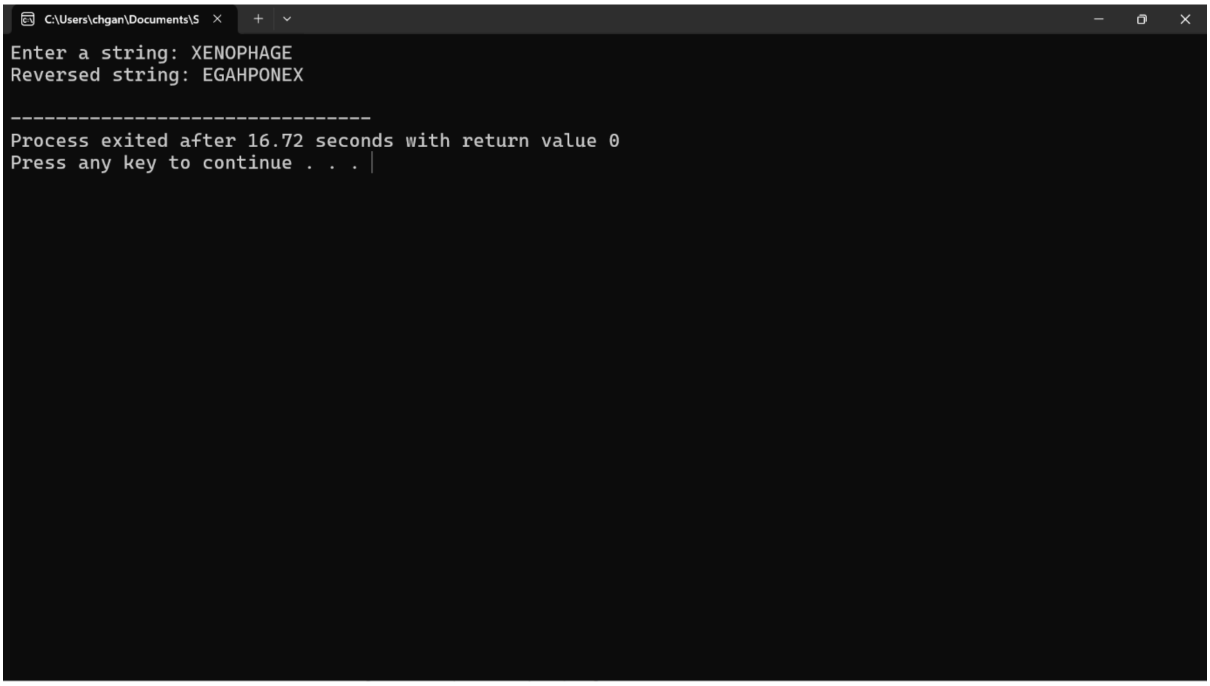
```
Enter number of elements: 4
Enter 4 sorted elements:
12 69 500 869
Enter the element to search: 69
Element found at index 1

--------------------------------
Process exited after 14.89 seconds with return value 0
Press any key to continue . . .
```

13. Program to perform the String reverse operation.

PROGRAM:

```
#include <stdio.h>
#include <string.h>
int main() {
    char str[100], temp;
    int length, i;
    printf("Enter a string: ");
    scanf("%s", str);
    length = strlen(str);
    for (i = 0; i < length / 2; i++) {
        temp = str[i];
        str[i] = str[length - i - 1];
        str[length - i - 1] = temp;
    }
    printf("Reversed string: %s\n", str);
    return 0;
}
```

OUTPUT:



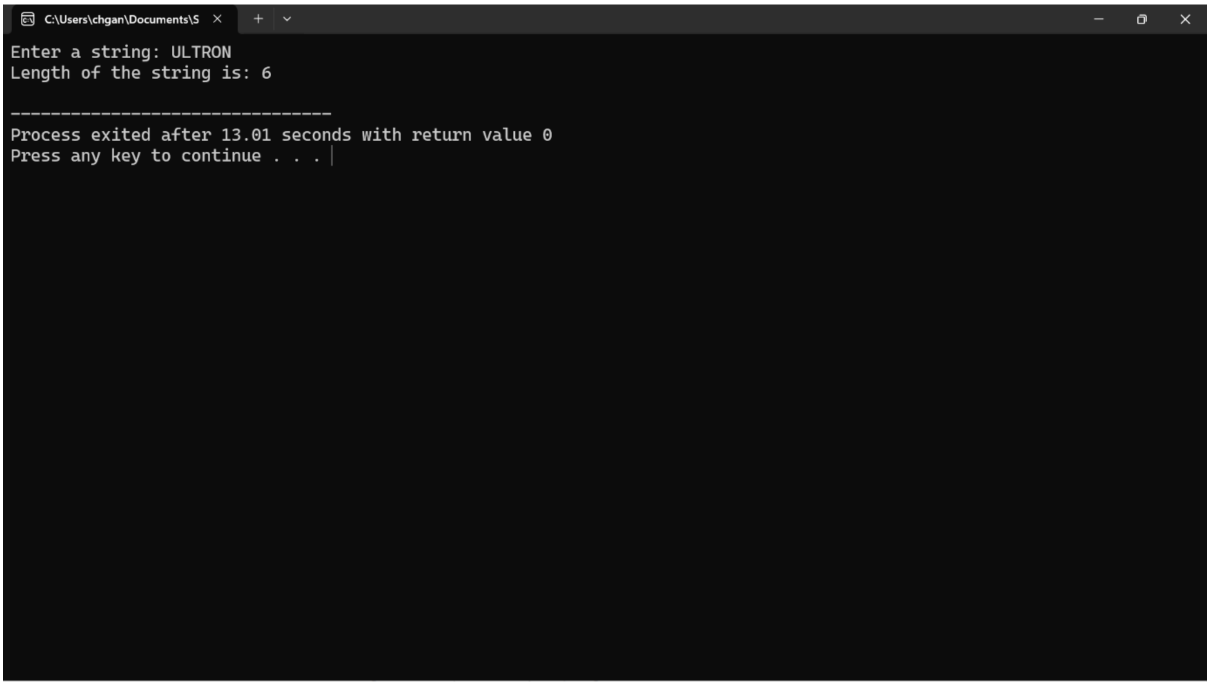
```
Enter a string: XENOPHAGE
Reversed string: EGAHPONEX

--------------------------------
Process exited after 16.72 seconds with return value 0
Press any key to continue . . .
```

14. Program to find the length of a String.

PROGRAM:

```
#include <stdio.h>
int main() {
    char str[100];
    int length = 0;
    printf("Enter a string: ");
    scanf("%s", str);
    while (str[length] != '\0') {
        length++;
    }
    printf("Length of the string is: %d\n", length);
    return 0;
}
```

OUTPUT:



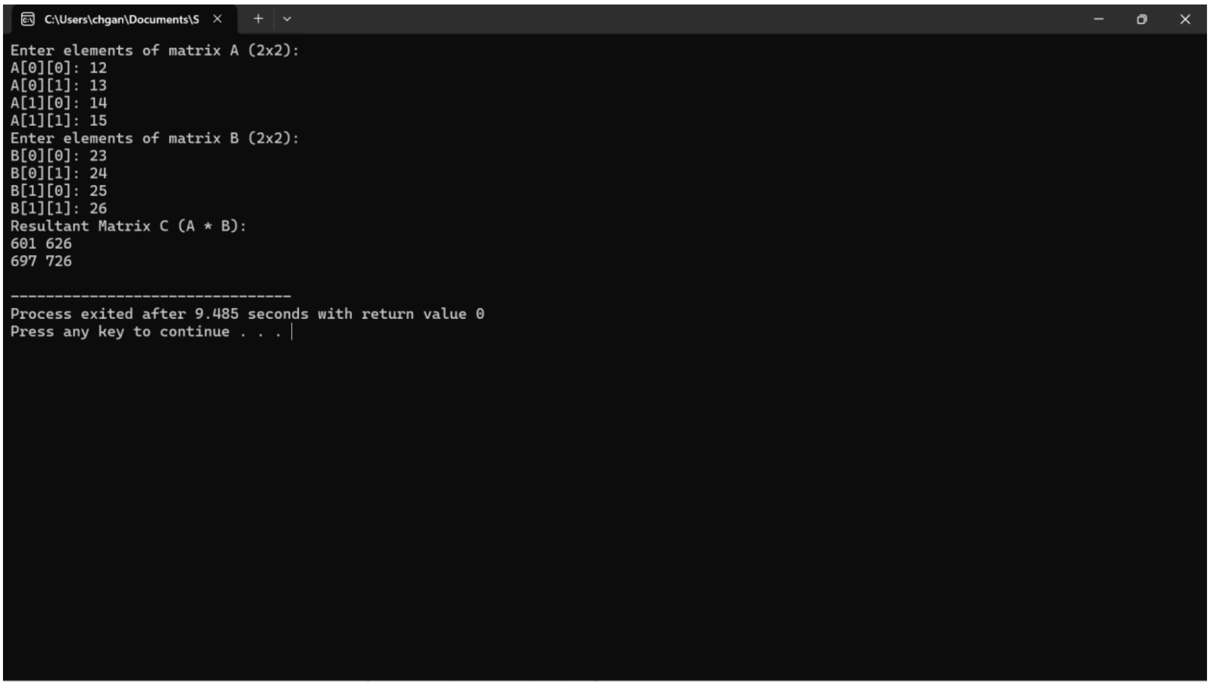
```
Enter a string: ULTRON
Length of the string is: 6

--------------------------------
Process exited after 13.01 seconds with return value 0
Press any key to continue . . .
```

15. Program to perform Strassen’s matrix multiplication.

PROGRAM:

```
#include <stdio.h>
void strassen(int A[2][2], int B[2][2], int C[2][2]) {
    int P1 = A[0][0] * (B[0][1] - B[1][1]);
    int P2 = (A[0][0] + A[0][1]) * B[1][1];
    int P3 = (A[1][0] + A[1][1]) * B[0][0];
    int P4 = A[1][1] * (B[1][0] - B[0][0]);
    int P5 = (A[0][0] + A[1][1]) * (B[0][0] + B[1][1]);
    int P6 = (A[0][1] - A[1][1]) * (B[1][0] + B[1][1]);
    int P7 = (A[0][0] - A[1][0]) * (B[0][0] + B[0][1]);
    C[0][0] = P5 + P4 - P2 + P6;
    C[0][1] = P1 + P2;
    C[1][0] = P3 + P4;
    C[1][1] = P5 + P1 - P3 - P7;
}
int main() {
    int A[2][2], B[2][2], C[2][2];
    printf("Enter elements of matrix A (2x2):\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("A[%d][%d]: ", i, j);
            scanf("%d", &A[i][j]);
        }
    }
    printf("Enter elements of matrix B (2x2):\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("B[%d][%d]: ", i, j);
            scanf("%d", &B[i][j]);
        }
    }
    strassen(A, B, C);
    printf("Resultant Matrix C (A * B):\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:



```
Enter elements of matrix A (2x2):
A[0][0]: 12
A[0][1]: 13
A[1][0]: 14
A[1][1]: 15
Enter elements of matrix B (2x2):
B[0][0]: 23
B[0][1]: 24
B[1][0]: 25
B[1][1]: 26
Resultant Matrix C (A * B):
601 626
697 726

--------------------------------
Process exited after 9.485 seconds with return value 0
Press any key to continue . . .
```