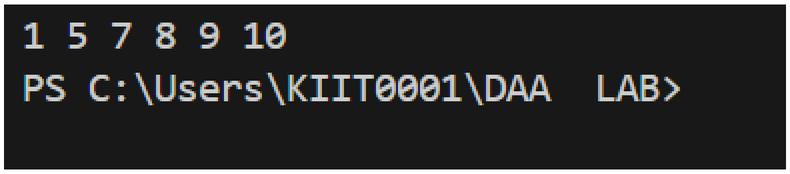
LAB TASK-6

1. Write a program to sort the array elements using Quick Sort Technique.

```c
#include <stdio.h>

void swap(int* a, int* b);

int partition(int arr[], int low, int high) {


    int pivot = arr[high];


    int i = low - 1;


    for (int j = low; j <= high - 1; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }


    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}


void quickSort(int arr[], int low, int high) {
    if (low < high) {


        int pi = partition(arr, low, high);


        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

void swap(int* a, int* b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int main() {
    int arr[] = {10, 7, 8, 9, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    quickSort(arr, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

OUTPUT

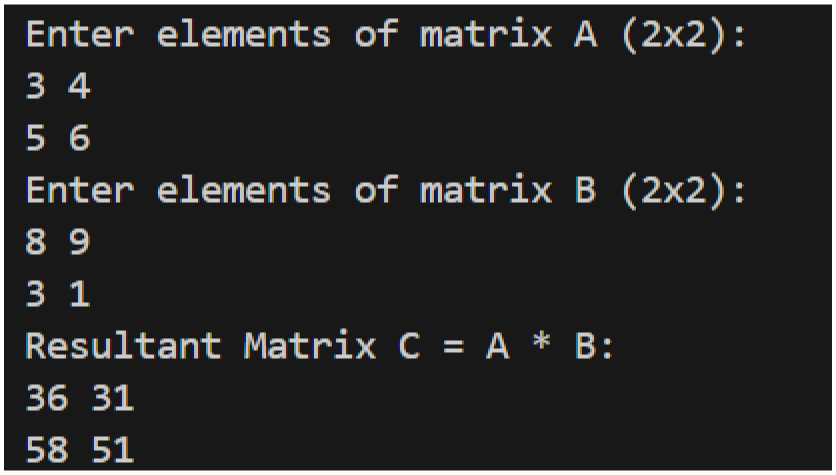


```
1 5 7 8 9 10
PS C:\Users\KIIT0001\DAA  LAB>
```

2. Divide and conquer: Implementation of Strassen's algorithm for matrix multiplication. Also analyze how this approach is advantage when compared to normal multiplication.

```c
#include <stdio.h>

// Function to implement Strassen’s Algorithm (only for 2x2 matrices here)
void strassen(int A[2][2], int B[2][2], int C[2][2]) {
    int M1, M2, M3, M4, M5, M6, M7;

    // Using Strassen's formulas
    M1 = (A[0][0] + A[1][1]) * (B[0][0] + B[1][1]);
    M2 = (A[1][0] + A[1][1]) * B[0][0];
    M3 = A[0][0] * (B[0][1] - B[1][1]);
    M4 = A[1][1] * (B[1][0] - B[0][0]);
    M5 = (A[0][0] + A[0][1]) * B[1][1];
    M6 = (A[1][0] - A[0][0]) * (B[0][0] + B[0][1]);
    M7 = (A[0][1] - A[1][1]) * (B[1][0] + B[1][1]);

    // Computing the result matrix C
    C[0][0] = M1 + M4 - M5 + M7;
    C[0][1] = M3 + M5;
    C[1][0] = M2 + M4;
    C[1][1] = M1 - M2 + M3 + M6;
}

int main() {
    int A[2][2], B[2][2], C[2][2];
    int i, j;

    printf("Enter elements of matrix A (2x2):\n");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf("%d", &A[i][j]);

    printf("Enter elements of matrix B (2x2):\n");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf("%d", &B[i][j]);

    // Perform multiplication using Strassen
    strassen(A, B, C);

    printf("Resultant Matrix C = A * B:\n");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++)
            printf("%d ", C[i][j]);
        printf("\n");
    }

    return 0;
}
```

OUTPUT



```
Enter elements of matrix A (2x2):
3 4
5 6
Enter elements of matrix B (2x2):
8 9
3 1
Resultant Matrix C = A * B:
36 31
58 51
```