DAA Practical No: 05

Q) Write a C/C++ code to implement (with practical example implementation)

1)Binary Search

2)Merge Sort

3)Quick Sort

4)Strassen's Matrix multiplication

1)Binary Search

-- >Program Code:

Go Run Terminal Help BinarySearch.cpp - DAA - Visual Studio Code

BinarySearch.cpp

p5 > BinarySearch.cpp > binary(int [], int, int)

```cpp
//Binary Search
#include<iostream>
using namespace std;

int binary(int arr[],int size,int key ){
    int start=0;
    int end=size-1;

    // int mid= (start + end)/2;   //if two big values of start and end we take
                                   // then it may go out of range of int
    int mid= start + (end-start)/2;


    while(start <= end){

        if(arr[mid] == key){
            return mid;
        }

        //to go right part
        else if(arr[mid]> key){
            start=mid+1;
        }

        //to go left part
        else {
            end = mid-1;
        }

        mid= start + (end-start)/2;
    }
    return -1;

}
```

```cpp
int main(){
    int arr[5]={1,2,5,10,20};
    int brr[6]={20,10,8,7,5,3};

    int key;
    cout<<"Enter number to search in array:"<<endl;
    cin>>key;


    cout<<"Index of key in oddlength array is:"<<binary(arr,5,key)<<endl;
    cout<<"index of key in evenlength array is:"<<binary(brr,6,key)<<endl;

}
```

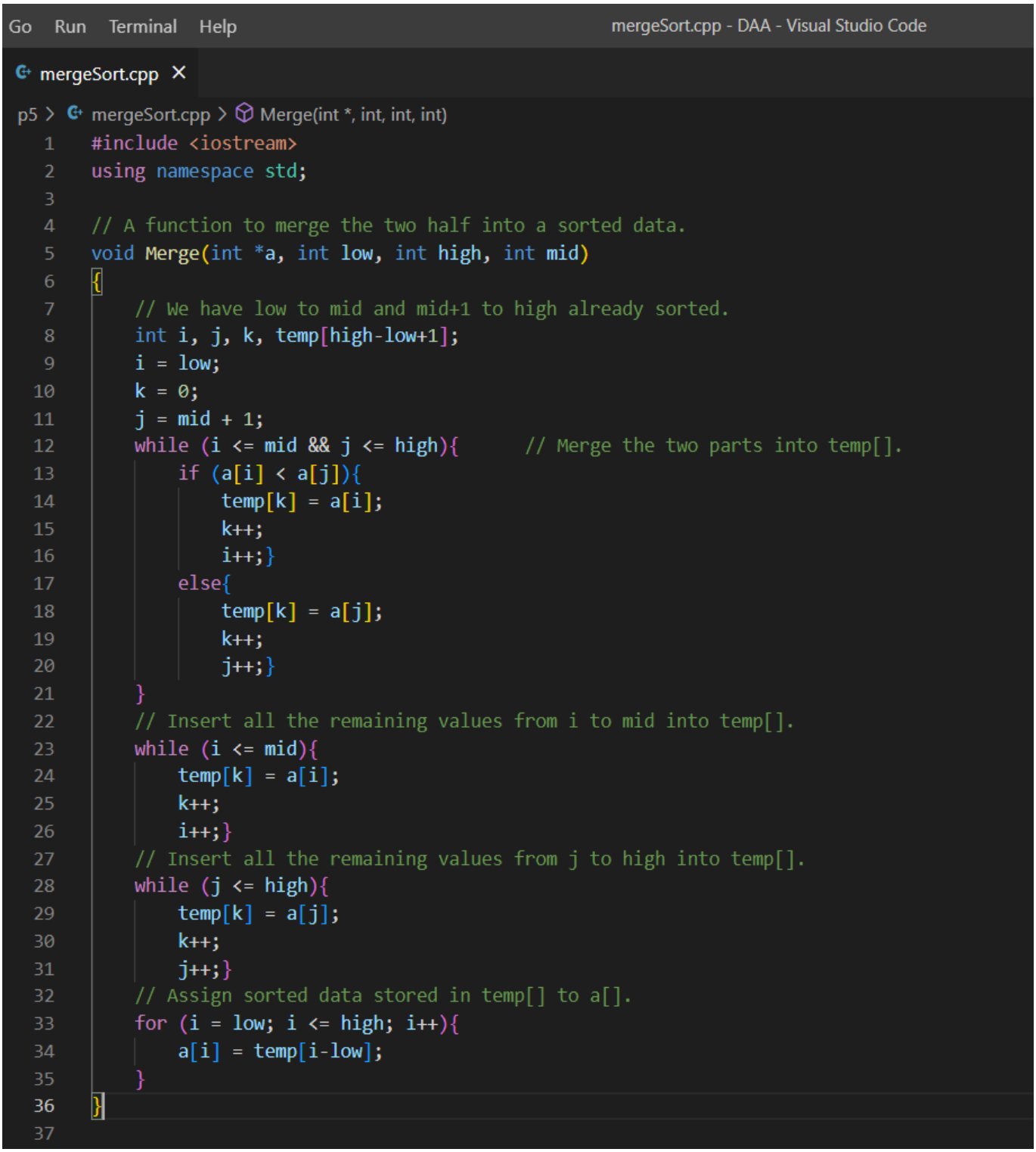
Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\DELL\Desktop\c++> cd "c:\Users\DELL\Desktop\c++"
PS C:\Users\DELL\Desktop\c++> & .\"22binarySearch.exe"
Enter number to search in array:
5
Index of key in oddlength array is:2
index of key in evenlength array is:4
PS C:\Users\DELL\Desktop\c++>
```

2)Merge Sort:

-- >Program Code:



Go Run Terminal Help mergeSort.cpp - DAA - Visual Studio Code

mergeSort.cpp

p5 > mergeSort.cpp > Merge(int *, int, int, int)

```cpp
// A function to split array into two parts.
void MergeSort(int *a, int low, int high)
{
    int mid;
    if (low < high){
        mid=(low+high)/2;
        // Split the data into two half.
        MergeSort(a, low, mid);
        MergeSort(a, mid+1, high);

        // Merge them to get sorted output.
        Merge(a, low, high, mid);
    }
}

int main(){
    int n, i;
    cout<<"\nEnter the number of data element to be sorted: ";
    cin>>n;

    int arr[n];
    for(i = 0; i < n; i++){
        cout<<"Enter element "<<i+1<<": ";
        cin>>arr[i];
    }

    MergeSort(arr, 0, n-1);

    // Printing the sorted data.
    cout<<"\nSorted Data ";
    for (i = 0; i < n; i++)
        cout<<"->"<<arr[i];

    return 0;
}
```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5"
PS C:\Users\DELL\Desktop\DAA\p5> & .\"mergeSort.exe"

Enter the number of data element to be sorted: 5
Enter element 1: 20
Enter element 2: 45
Enter element 3: 85
Enter element 4: 10
Enter element 5: 0

Sorted Data ->0->10->20->45->85
PS C:\Users\DELL\Desktop\DAA\p5>
```

3)Quick Sort:

-- > Program Code:

Go Run Terminal Help quickSort.cpp - DAA - Visual Studio Code

quickSort.cpp

p5 > quickSort.cpp > ...

```cpp
//Quick Sort
#include<iostream>
using namespace std;

void swap(int arr[] , int pos1, int pos2){
    int temp;
    temp = arr[pos1];
    arr[pos1] = arr[pos2];
    arr[pos2] = temp;
}

int partition(int arr[], int low, int high, int pivot){
    int i = low;
    int j = low;
    while( i <= high){
        if(arr[i] > pivot){
            i++;
        }
        else{
            swap(arr,i,j);
            i++;
            j++;
        }
    }
    return j-1;
}

void quickSort(int arr[], int low, int high){
    if(low < high){
    int pivot = arr[high];
    int pos = partition(arr, low, high, pivot);

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

int main()
{
    int n ;
    cout << " enter the size of array: ";
    cin>>n;
    int arr[n];
    cout<<"Enter elements in array:";
    for( int i = 0 ; i < n; i++){
        cin>> arr[i];
    }
    quickSort(arr, 0 , n-1);
    cout<<"The sorted array is: "<<endl;
    for( int i = 0 ; i < n; i++){
        cout<< arr[i]<<" ";
    }

}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5\output"
PS C:\Users\DELL\Desktop\DAA\p5\output> & .\"quickSort.exe"
 enter the size of array: 5
Enter elements in array:45 10 20 80 70
The sorted array is:
10 20 45 70 80
PS C:\Users\DELL\Desktop\DAA\p5\output>
```

4)Strassen's Matrix multiplication

Program Code:

Go Run Terminal Help matrix_multiplication.cpp - DAA - Visual Studio Code

matrix_multiplication.cpp

p5 > matrix_multiplication.cpp > main()

```cpp
#include <iostream>
using namespace std;

void multiply(int[5][5], int[5][5], int, int, int);
int display(int[5][5], int, int);

int main()
{
    int a[5][5], b[5][5], r1, c1, r2, c2;
    cout << "\n Enter rows for first matrix: ";
    cin >> r1;
    cout << "\n Enter columns for first matrix: ";
    cin >> c1;

    cout << "\n Enter rows for second matrix: ";
    cin >> r2;
    cout << "\n Enter columns for second matrix: ";
    cin >> c2;

    // To check if columns of first matrix are equal to rows of second matrix
    if (c1 != r2)
        return 0;
    // Storing elements of first matrix.

    cout << "\n Enter elements of first matrix \n";

    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c1; j++)
            cin >> a[i][j];
    }
    // Storing elements of second matrix.
    cout << "\n Enter elements of second matrix\n";

    for (int i = 0; i < r2; i++) {
        for (int j = 0; j < c2; j++)
            cin >> b[i][j];
    }
```

```cpp
    display(a, r1, c1);
    display(b, r2, c2);
    //calling the function to multiply a and b. passing number of rows
    //and columns in both of them
    multiply(a, b, r1, c2, c1);
    return 0;
}
void multiply(int a[5][5], int b[5][5], int row, int col, int c1)
{
    int c[5][5];
    //input 0 for all values of c, in order to remove
    //the garbage values assigned earlier
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
            c[i][j] = 0;
    }
    //we apply the same formula as above
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            for (int k = 0; k < c1; k++) //columns of first matrix || rows of second matrix
                c[i][j] += a[i][k] * b[k][j];
        }
    }
    cout << "\n Matrix c after matrix multiplication is:\n";
    display(c, row, col);
}
int display(int c[5][5], int row, int col)
{
    cout << "\n Matrix is:\n";
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
            cout << c[i][j] << " ";
        cout << "\n";
    }
    return 0;
}
```

Output:

```
Go  Run  Terminal  Help                    matrix_multiplication.cpp - DAA - Visual Studio Code

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\DELL\Desktop\DAA> cd "c:\Users\DELL\Desktop\DAA\p5\output"
PS C:\Users\DELL\Desktop\DAA\p5\output> & .\"matrix_multiplication.exe"

 Enter rows for first matrix: 2

 Enter columns for first matrix: 3

 Enter rows for second matrix: 3

 Enter columns for second matrix: 2

 Enter elements of first matrix
5 8 9
1 3 7

 Enter elements of second matrix
5 2
10 4
7 3

 Matrix is:
5 8 9
1 3 7

 Matrix is:
5 2
10 4
7 3

 Matrix c after matrix multiplication is:

 Matrix is:
168 69
84 35
PS C:\Users\DELL\Desktop\DAA\p5\output>
```