

```

//Linear Search

#include <iostream>
using namespace std;
class linear
{
    int *a;

public:
    int n;
    void read()
    {
        cout << "Enter the size of the array :\n";
        cin >> n;
        a = new int[n];
        cout << "Enter the elements of the array:\n";
        for (int i = 0; i < n; i++)
        {
            cin >> a[i];
        }
    }
    void search()
    {
        int item, i;
        int comp=0; // for storing no of comparisons
        cout << "Enter the item to be searched:\n";
        cin >> item;
        for (i = 0; i < n; i++)
        {
            comp++;
            if (a[i] == item)
            {
                cout << item << " found! at index " <<
i << "\n";
                cout << "No of comparisons : " << comp
<< endl;
                return;
            }
        }
    }
}

```

```

        }
        if (i == n)
        {
            cout << item << " is not found in the
array!\n";
        }
    }
    ~linear()
    {
        delete []a;
    }
};
int main()
{
    linear a;
    a.read();
    a.search();
    return 0;
}

```

//Binary Search

```

#include <iostream>
using namespace std;
class binary
{
    int *a;

public:
    int n;
    void read()
    {
        cout << "Enter the size of the array :\n";
        cin >> n;
        a = new int[n];
        cout << "Enter the elements of the array:\n";
        for (int i = 0; i < n; i++)
        {

```

```

        cin >> a[i];
    }
}
void sort()
{
    for(int i = 0; i < n - 1; i++)
    {
        for(int j = i + 1; j < n; j++)
        {
            if(a[i] > a[j])
            {
                int temp = a[j];
                a[j] = a[i];
                a[i] = temp;
            }
        }
    }
}
void search()
{
    int item, comp = 0;
    cout << "Enter the item to be searched:\n";
    cin >> item;
    int start = 0, end = n - 1;
    while (start <= end)
    {
        comp++;
        int mid = (start + end) / 2;
        if (a[mid] == item)
        {
            cout << item << " is found " << endl;
            cout << "no of comparisons :
" << comp << endl;
            return;
        }
        if (item < a[mid])
        {
            end = mid - 1;

```

```

        }
        else
        {
            start = mid + 1;
        }
    }
    cout<<item<<" is not found in the array!\n";
}
~binary()
{
    delete []a;
}
};
int main()
{
    binary a;
    a.read();
    a.sort();
    a.search();
    return 0;
}

```

//Insertion Sort

```

#include <iostream>
using namespace std;
class isort
{
    int *a;

public:
    int n;
    void read()
    {
        cout << "Enter the size of the array :\n";
        cin >> n;
        a = new int[n];
        cout << "Enter the elements of the array:\n";
    }
}

```



```
    a.sort();
    a.display();
    return 0;
}
```

//Bubble Sort

```
#include <iostream>
using namespace std;
class bubble
{
    int *a;
public:
    int n;
    void read()
    {
        cout << "Enter the size of the array :\n";
        cin >> n;
        a = new int[n];
        cout << "Enter the elements of the array:\n";
        for (int i = 0; i < n; i++)
        {
            cin >> a[i];
        }
    }
    void display()
    {
        cout << "The elements in the array are:\n";
        for (int i = 0; i < n; i++)
        {
            cout<<a[i]<<"\t";
        }
        cout<<endl;
    }
    void sort()
    {
        for(int i = 0;i<n-1;i++)
        {
```

```

        for(int j = 0; j < n-i-1; j++)
        {
            if(a[j+1] < a[j])
            {
                int temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}
~bubble()
{
    delete []a;
}
};
int main()
{
    bubble a;
    a.read();
    a.display();
    a.sort();
    a.display();
    return 0;
}

```

//Quick Sort

```

#include <iostream>
using namespace std;
class sortq
{
    int *a;
public:
    int n, count = 0;
    void read()
    {
        cout << "Enter the size of the array :\n";
    }
}

```

```

    cin >> n;
    a = new int[n];
    cout << "Enter the elements of the array:\n";
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
}
void display()
{
    cout << "The elements in the array are:\n";
    for (int i = 0; i < n; i++)
    {
        cout<<a[i]<<"\t";
    }
    cout<<endl;
}
int partition(int l,int u)
{
    int pivot = a[u];
    int i = l-1;
    for(int j = l;j<=u;j++)
    {
        if(a[j]<pivot)//for descending order put
if(a[j]>pivot)
        {
            i++;
            int temp = a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
    i++;
    int temp = a[i];
    a[i]=a[u];
    a[u]=temp;
    return i;
}

```



```

void sort(int l,int u)
{
    if(l<=u)
    {
        int p = partition(l,u);
        sort(l,p-1);
        sort(p+1,u);
    }
}
void sort()
{
    sort(0,n-1);
}
~sortq()
{
    delete []a;
}
};
int main()
{
    sortq a;
    a.read();
    a.display();
    a.sort();
    a.display();
    return 0;
}

```

//Polynomial Addition

```

#include <iostream>
using namespace std;
class poly
{
    int coeff[10], exp[10], n;

public:
    void read()

```

```

{
    cout << "Enter the no of terms in the
polynomial:\n";
    cin >> n;
    cout << "Enter the polynomial:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "\tEnter the coefficient: ";
        cin >> coeff[i];
        cout << "\tEnter the exponent: ";
        cin >> exp[i];
    }
}

void display()
{
    for (int i = 0; i < n; i++)
    {
        if (exp[i] != 0)
        {
            cout << coeff[i] << " x^" << exp[i];
        }
        else
            cout << coeff[i];
        if (i != n - 1)
            cout << " + ";
    }
    cout << endl;
}

poly operator+(poly a)
{
    poly b;
    int i = 0, j = 0, k = 0;
    while (i < n && j < a.n)
    {
        if (exp[i] == a.exp[j])
        {
            b.coeff[k] = coeff[i] + a.coeff[j];
            b.exp[k] = exp[i];

```

```

        i++;
        j++;
    }
    else if (exp[i] > a.exp[j])
    {
        b.coeff[k] = coeff[i];
        b.exp[k] = exp[i];
        i++;
    }
    else
    {
        b.coeff[k] = a.coeff[j];
        b.exp[k] = a.exp[j];
        j++;
    }
    k++;
}
while (i < n)
{
    b.coeff[k] = coeff[i];
    b.exp[k] = exp[i];
    i++;
    k++;
}
while (j < a.n)
{
    b.coeff[k] = a.coeff[j];
    b.exp[k] = a.exp[j];
    j++;
    k++;
}
b.n = k;
return b;
}
};
int main()
{
    poly a, b, c;

```

```

    cout << "Enter the first polynomial:\n";
    a.read();
    cout << "Enter the second polynomial:\n";
    b.read();
    cout<<"The first polynomial:\n";
    a.display();
    cout<<"The second polynomial:\n";
    b.display();
    c = a + b;
    cout << "Sum is :\n";
    c.display();
    return 0;
}

```

//Stack Operations

```

#include <iostream>
using namespace std;
class stack
{
    int a[50],n,top;
public:
    stack()
    {
        cout<<"Enter the size of the stack:\n";
        cin>>n;
        top = -1;
    }
    void push(int item)
    {
        if(top == n-1)
        {
            cout<<"Stack Overflow!\n";
            return;
        }
        a[++top]=item;
        cout<<item<<" pushed into the stack\n";
    }
}

```

```

int pop()
{
    int data;
    if(top == -1)
    {
        cout<<"Stack underflow!\n";
        return NULL;
    }
    data = a[top];
    top--;
    return data;
}
void display()
{
    if(top == -1)
    {
        cout<<"Stack is empty!\n";
        return;
    }
    cout<<"***DISPLAYING STACK***\n\n";
    for(int i = top;i>=0;i--)
    {
        cout<<a[i]<<"\t";
    }
    cout<<endl<<endl;
}
};
int main()
{
    stack a;
    int op,n;
    char ch;
    do
    {
        cout<<"***MENU***\nEnter your
choice\n1)Push\n2)Pop\n3)Display\n";
        cin>>op;
        switch (op)

```

```

    {
        case 1:cout<<"Enter the item to be pushed in
to the stack:\n";
            cin>>n;
            a.push(n);
            break;
        case 2:
            int item;
            item = a.pop();
            if(item == NULL)
            {
                cout<<"Stack Empty!\n";
                break;
            }
            cout<<item<<" Was popped from the
stack\n";
            break;
        case 3:
            a.display();
            break;

        default:
            break;
    }
    cout<<"Do you want to continue (y/n):\n";
    cin>>ch;
} while (ch == 'y' || ch == 'Y');
return 0;
}

```

//Queue Operations

```

#include <iostream>
using namespace std;
class queue
{
    int a[50],n,front,rear;
public:

```

```
queue()
{
    cout<<"Enter the size of the queue:\n";
    cin>>n;
    front = -1;
    rear = -1;
}
void ins()
{
    int item;
    if(rear == n-1)
    {
        cout<<"Queue is full!\n";
        return;
    }
    cout<<"Enter the element to be inserted:\n";
    cin>>item;
    a[++rear]=item;
    if(front == -1)
        front = 0;
}
int del()
{
    int item;
    if(front == -1)
    {
        cout<<"Queue is empty!\n";
        return NULL;
    }
    item = a[front];
    if(front == rear)
    {
        front = -1;
        rear = -1;
    }
    else
        front++;
    return item;
}
```

```

    }
    void display()
    {
        if(front == -1)
        {
            cout<<"Queue is empty!\n";
            return;
        }
        cout<<"***Displaying Queue***\n\n";
        for(int i = front;i<=rear;i++)
        {
            cout<<a[i]<<"\t";
        }
        cout<<endl<<endl;
    }

};

int main()
{
    int op,item;
    char ch;
    queue a;
    do
    {
        cout<<"***Menu***\nEnter your
choice\n1)Insert\n2)Delete\n3)Display\n";
        cin>>op;
        switch (op)
        {
            case 1:
                a.ins();
                break;
            case 2:
                item = a.del();
                if(item == NULL)
                    break;
                else

```


