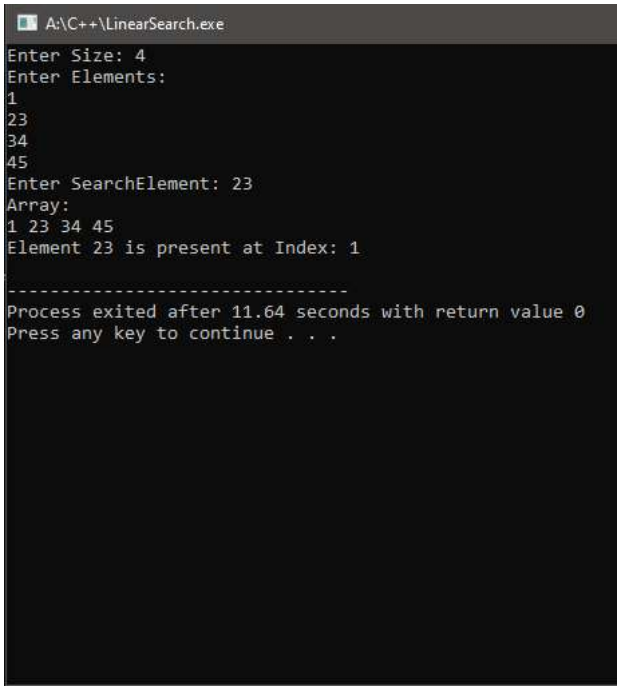
Week7 Batch3

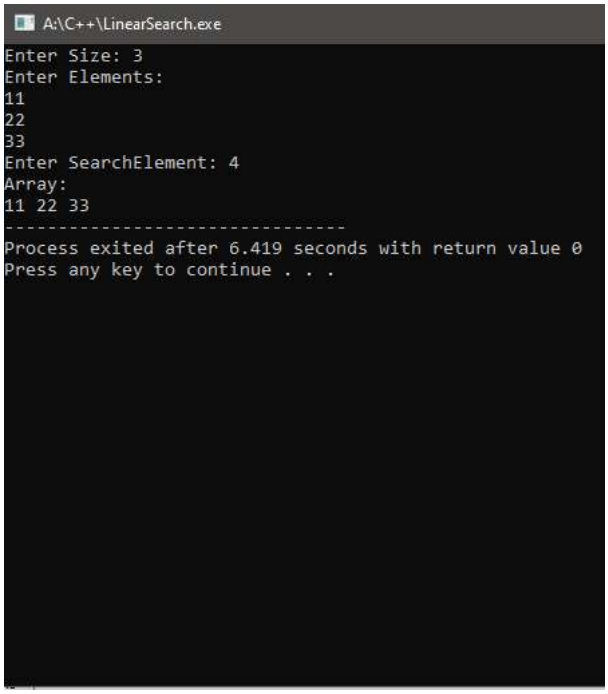
Saturday, January 01, 2022 1:57 PM

1. Linear Search and Binary searching

Linear Search

```
#include <bits/stdc++.h>
using namespace std;

void LinearSearch(int arr[], int n, int num)
{
   bool flag = true;
   for (int i = 0; i < n; i++)
   {
      if (arr[i] == num)
      {
         cout << "\nElement " << num << " is present at Index: " << i <<endl;
         flag = true;
         break;
      }
      else
         flag = false;
   }
   if (flag = false)
      cout << "\nElement " << num << " is not present" << endl;
}
void printArray(int arr[], int n)
{
   for (int i = 0; i < n; i++)
   {
      cout << arr[i] << " ";
   }
}

int main()
{
   int n, num;
   cout << "Enter Size: ";
   cin >> n;

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

   cout << "Enter SearchElement: ";
   cin >> num;

   cout << "Array: " << endl;
   printArray(arr, n);

   LinearSearch(arr, n, num);
```

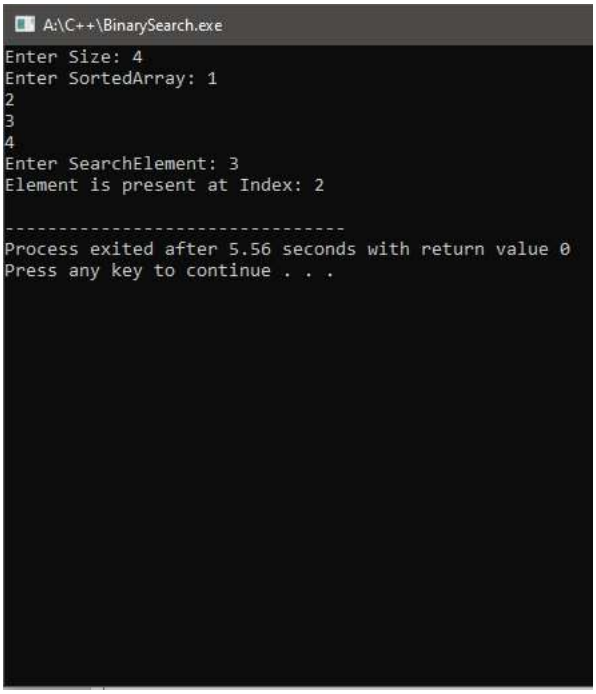


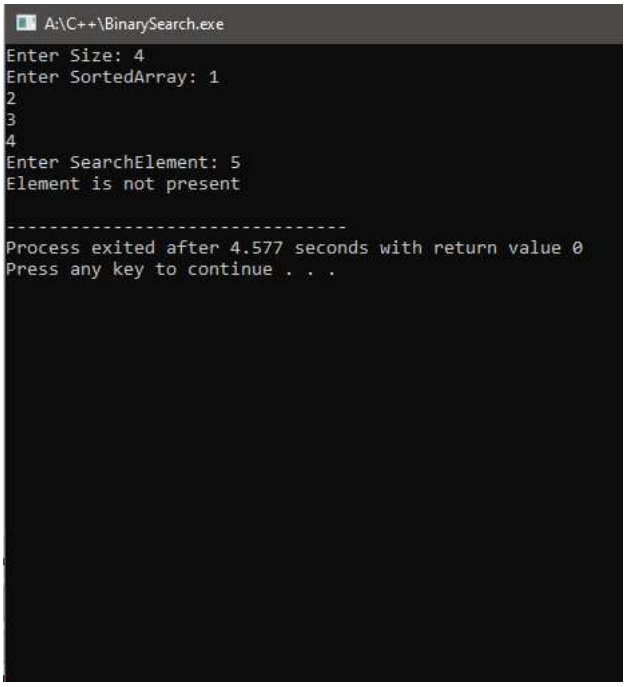


```
    return 0;
  }
```

Binary Search

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
   int n;
   int mid, low, high;

   cout << "Enter Size: ";
   cin >> n;
   int arr[n];
   cout << "Enter SortedArray: ";
   for (int i = 0; i < n; i++)
      cin >> arr[i];

   int key;
   cout << "Enter SearchElement: ";
   cin >> key;

   low = 0;
   high = n - 1;

   do
   {
      mid = (low + high) / 2;
      if (key < arr[mid]) // key in left half of sorted, so make new high as mid-1
         high = mid - 1;
      else if (key > arr[mid])
         low = mid + 1; // key in right half so make new low as mid+1
   } while (key != arr[mid] && low <= high);

   if (key == arr[mid])
   {
      cout << "Element is present at Index: " << mid << endl;
   }
   else
   {
      cout << "Element is not present\n";
   }
   return 0;
}
```





2. Sorting: Bubble, Quick, Selection & Insertion

BubbleSort

```
#include<iostream>
using namespace std;
int main()
{
   int n, i, arr[100], j, temp;
   cout<<"Enter Size: "<<endl;
   cin>>n;
   cout<<"Enter Elements: "<<endl;
   for(i=0; i<n; i++)
      cin>>arr[i];

   for(i=0; i<(n-1); i++)
   {
      for(j=0; j<(n-i-1); j++)
      {
         if(arr[j]>arr[j+1])
         {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
         }
      }
   }

   cout<<"SortedArray: "<<endl;
   for(i=0; i<n; i++)
      cout<<arr[i]<<" ";
   cout<<endl;
   return 0;
}
```

```
A:\C++\BubbleSort.exe
Enter Size:
5
Enter Elements:
34
12
54
23
75
SortedArray:
12 23 34 54 75

--------------------------------
Process exited after 9.379 seconds with return value 0
Press any key to continue . . .
```

QuickSort

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

int Partition(int arr[], int s, int e) {
int pivot = arr[e];
int pIndex = s;
for(int i = s;i<e;i++) {
if(arr[i]<pivot) {
int temp = arr[i];
arr[i] = arr[pIndex];
arr[pIndex] = temp;
pIndex++;
}
}
int temp = arr[e];
arr[e] = arr[pIndex];
arr[pIndex] = temp;
return pIndex;
}

void QuickSort(int arr[], int s, int e) {
```

```
A:\C++\QuickSort.exe
Enter Size:
5
Enter Elements:
123
345
1243
3564
1234
SortedArray:
123 345 1234 1243 3564
--------------------------------
Process exited after 6.749 seconds with return value 0
Press any key to continue . . .
```

```
if(s<e) {
int p = Partition(arr,s, e);
QuickSort(arr, s, (p-1));
QuickSort(arr, (p+1), e);
}
}

int main() {
int size=0;
cout<<"Enter Size: "<<endl;
cin>>size;
int arr[size];

cout<<"Enter Elements: "<<endl;
for(int i=0;i<size;i++) {
cin>>arr[i];
}

QuickSort(arr,0,(size-1));

cout<<"SortedArray: "<<endl;
for(int i=0;i<size;i++) {
cout<<arr[i]<<" ";
}
return 0;
}
```

SelectionSort

```
#include<iostream>
using namespace std;
int main()
{
   int s, arr[100], i, j, temp, small, c, index;
   cout<<"Enter Size: "<<endl;
   cin>>s;
   cout<<"Enter Elements: "<<endl;
   for(i=0; i<s; i++)
      cin>>arr[i];
   for(i=0; i<(s-1); i++)
   {
      c=0;
      small = arr[i];
      for(j=(i+1); j<s; j++)
      {
         if(small>arr[j])
         {
            small = arr[j];
            c++;
            index = j;
         }
      }
      if(c!=0)
      {
         temp = arr[i];
         arr[i] = small;
```

```
            arr[index] = temp;
        }
    }
    cout<<"SortedArray: "<<endl;
    for(i=0; i<s; i++)
        cout<<arr[i]<<" ";
    cout<<endl;
    return 0;
}
```

InsertionSort

```
#include<iostream>
using namespace std;
int main()
{
    int arr[100], s, i, j, k, e, index;
    cout<<"Enter Size: "<<endl;
    cin>>s;
    cout<<"Enter Elements: "<<endl;
    for(i=0; i<s; i++)
        cin>>arr[i];
    for(i=1; i<s; i++)
    {
        e = arr[i];
        if(e<arr[i-1])
        {
            for(j=0; j<=i; j++)
            {
                if(e<arr[j])
                {
                    index = j;
                    for(k=i; k>j; k--)
                        arr[k] = arr[k-1];
                    break;
                }
            }
        }
        else
            continue;
        arr[index] = e;
    }
    cout<<"SortedArray: "<<endl;
    for(i=0; i<s; i++)
        cout<<arr[i]<<"  ";
    cout<<endl;
    return 0;
}
```