DAA file - DAA file for AKTU Students in Btech CSE 3rd Year

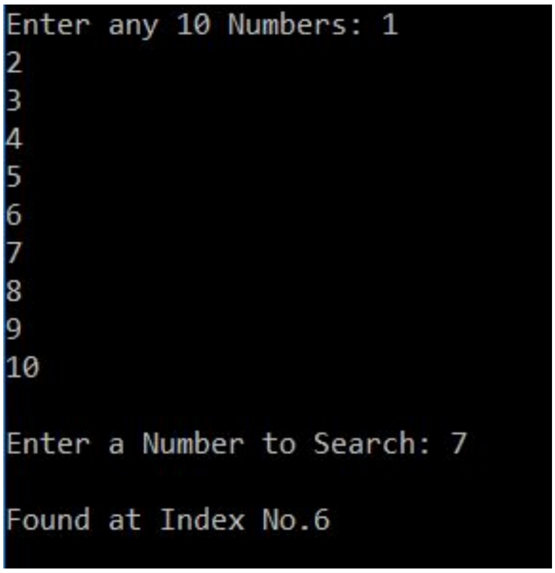
B.tech (Dr. A.P.J. Abdul Kalam Technical University)

Program 1: - Write a program in C++ to perform linear Search.

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int arr[10], i, num, index;
    cout<<"Enter 10 Numbers: ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<"\nEnter a Number to Search: ";
    cin>>num;
    for(i=0; i<10; i++)
    {
        if(arr[i]==num)
        {
            index = i;
            break;
        }
    }
    cout<<"\nFound at Index No."<<index;
    cout<<endl;
    return 0;
}
```

Output:



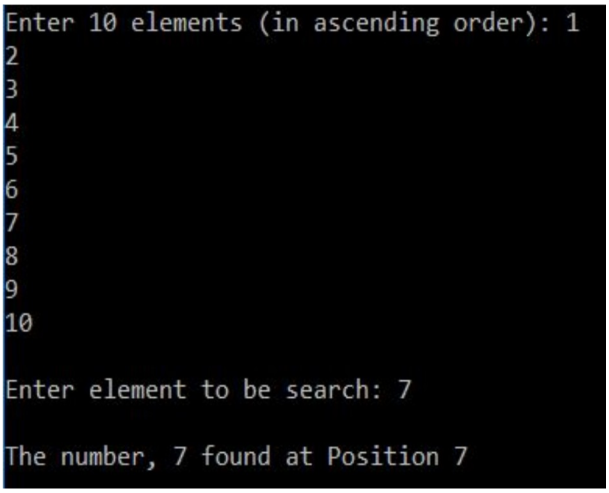
```
Enter any 10 Numbers: 1
2
3
4
5
6
7
8
9
10

Enter a Number to Search: 7

Found at Index No.6
```

Program 2: - Write a program in C++ to perform Binary Search.

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int i, arr[10], num, first, last, middle;
    cout<<"Enter 10 Elements (in ascending order): ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<"\nEnter Element to be Search: ";
    cin>>num;
    first = 0;
    last = 9;
    middle = (first+last)/2;
    while(first <= last)
    {
        if(arr[middle]<num)
            first = middle+1;
        else if(arr[middle]==num)
        {
            cout<<"\nThe number, "<<num<<" found at Position "<<middle+1;
            break;
        }
        else
            last = middle-1;
        middle = (first+last)/2;
    }
    if(first>last)
        cout<<"\nThe number, "<<num<<" is not found in given Array";
    cout<<endl;
    return 0;
}
```

Output:



```
Enter 10 elements (in ascending order): 1
2
3
4
5
6
7
8
9
10

Enter element to be search: 7

The number, 7 found at Position 7
```

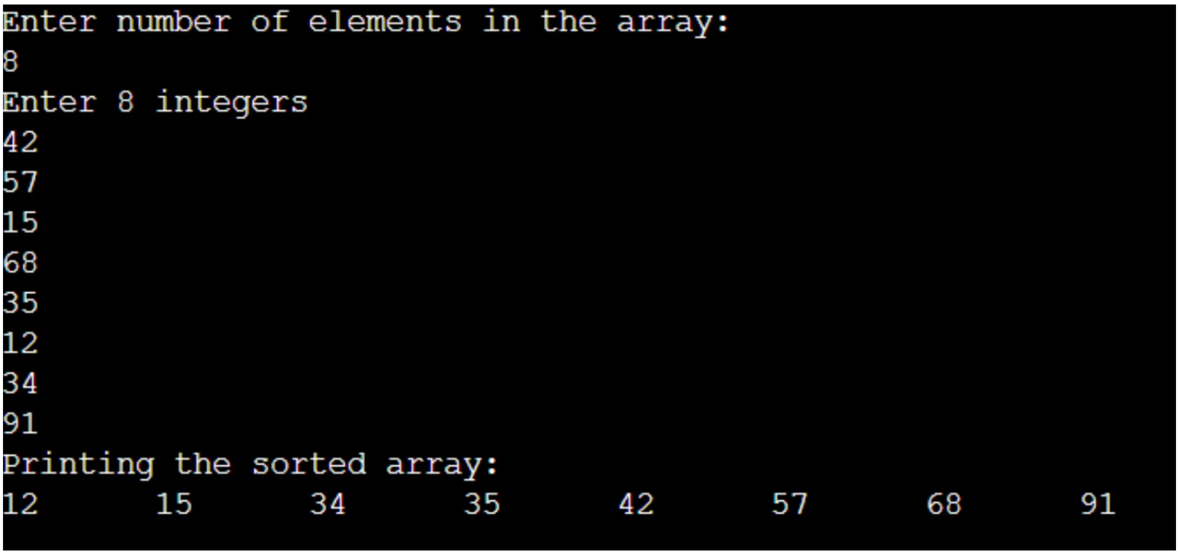
Program 3: - Write a program in C++ to perform Merge Sort.

Code:

```
#include<iostream.h>
using namespace std;
void mergeSort(int a[], int Firstindex, int Lastindex)
{ if (Firstindex < Lastindex)
   { int m = Firstindex + (Lastindex - Firstindex)/2;
     mergeSort(a, Firstindex, m);
      mergeSort(a, m+1, Lastindex);
      merge(a, Firstindex, m, Lastindex);
   }}
void merge(int a[], int Firstindex, int m, int Lastindex)
{  int x, y , z, sub1 = m - Firstindex + 1, sub2 =  Lastindex - m;
   int First[sub1], Second[sub2];
   for (x = 0; x < sub1; x++)
      First[x] = a[Firstindex + x];
   for (y = 0; y < sub2; y++)
      Second[y] = a[m + 1+ y];
   x = 0,y=0, z = Firstindex;
   while (x < sub1 && y < sub2)
   { if (First[x] <= Second[y])
      {   a[z] = First[x];
         x++;} else {
         a[z] = Second[y];
         y++;
      }z++;}
   while (x < sub1)
   { a[z] = First[x];
      x++;
```

```
        z++;}
    while (y < sub2)
    { a[z] = Second[y];
        y++; z++;  } }
int main()
{   int size;
    cout<<"Enter number of elements in the Array: ";
cin>>size;
    int Hello[size],i;
    cout<<"Enter " <<size<<"elements \n";
    for(i=0; i<size; i++)
             cin>>Hello[i];
    mergeSort(Hello, 0, size - 1);
    cout<<"The Sorted List isn";
    for(i=0; i<size; i++)
    { cout<<Hello[i]<<" ";}
    return 0;}
```

Output:



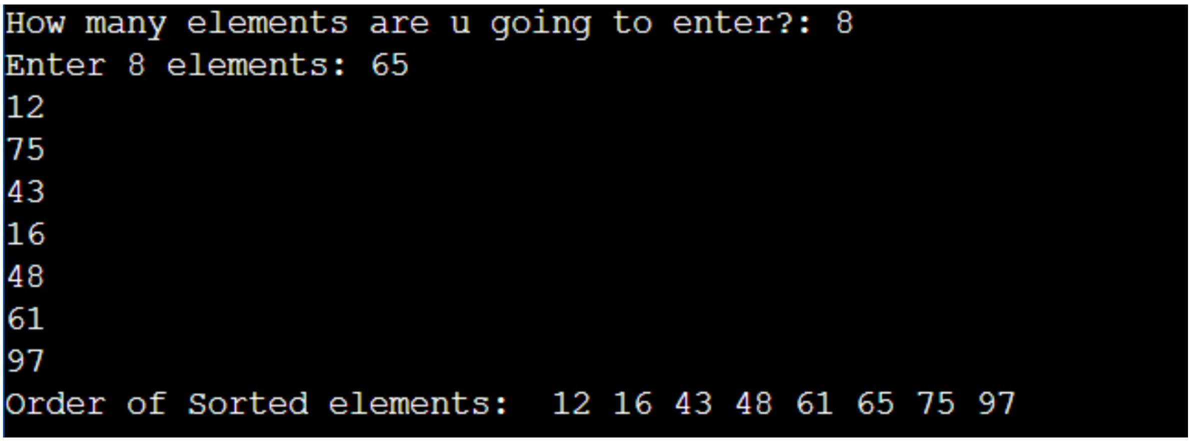
```
Enter number of elements in the array:
8
Enter 8 integers
42
57
15
68
35
12
34
91
Printing the sorted array:
12      15      34      35      42      57      68      91
```

Program 4: - Write a program in C++ to perform Quick Sort.

Code:

```
#include<iostream>
using namespace std;
void QUICKSORT(int [],int ,int );
int PARTITION(int [],int,int );
int main()
{    int n;
    cout<<" How many elements are you going to enter?:  "<<endl;
    cin>>n;
    int a[n];
    cout<<"Enter "<<n<<" Elements: ";
    for(int i=1;i<=n;i++) {
        cin>>a[i];   }
 int p=1,r=n;
    QUICKSORT(a,p,r);
    cout<<"Order of sorted elements"<<endl;
   for(int i=1;i<=n;i++)
   {cout<<a[i]<<”  ”;}
return 0;}
void QUICKSORT(int a[],int p,int r)
    {   int q;
        if(p<r)
        {q=PARTITION(a,p,r);
         QUICKSORT(a,p,q-1);
         QUICKSORT(a,q+1,r);}}
 int PARTITION(int a[],int p,int r)
    {int temp,temp1;
        int x=a[r], i=p-1;
        for(int j=p;j<=r-1;j++)
        {   if(a[j]<=x)
            {i=i+1;
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;}}
        temp1=a[i+1];
        a[i+1]=a[r];
        a[r]=temp1;
        return i+1;}
```

Output:



```
How many elements are u going to enter?: 8
Enter 8 elements: 65
12
75
43
16
48
61
97
Order of Sorted elements:  12 16 43 48 61 65 75 97
```

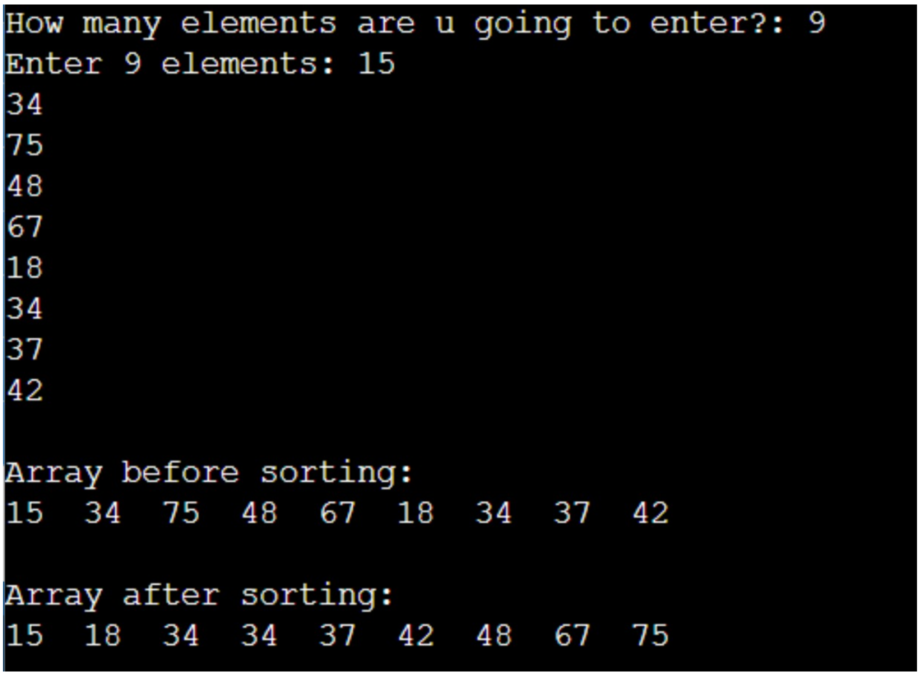
Program 5: - Write a program in C++ to perform Heap Sort.

Code:

```
 #include <iostream>

 using namespace std;

 void heapify(int arr[], int n, int i) {

   int largest = I, left = 2 * i + 1, right = 2 * i + 2;

   if (left < n && arr[left] > arr[largest])

     largest = left;

   if (right < n && arr[right] > arr[largest])

     largest = right;

   if (largest != i) {

     swap(arr[i], arr[largest]);

     heapify(arr, n, largest);

   }}

void heapSort(int arr[], int n) {

 for (int i = n / 2 - 1; i >= 0; i--)

     heapify(arr, n, i);

 for (int i = n - 1; i >= 0; i--) {

     swap(arr[0], arr[i]);

 heapify(arr, i, 0);}}

void printArray(int arr[], int n) {

   for (int i = 0; i < n; ++i)

     cout << arr[i] << " ";}

int main() {

   int arr[50], n;

   cout<<"How many elements are you going to enter?:  ";
   cin>>n;
   cout<<"Enter "<<n<<" Elements: ";
   for(i=0; i<n; i++)
       cin>>arr[i];
   heapSort(arr, n);

   cout << "Array before sorting: \n";

   printArray(arr, n);

   heapSort(arr, n);

   cout << "Array after sorting: \n";
```

```
    printArray(arr, n);  }
```

Output:



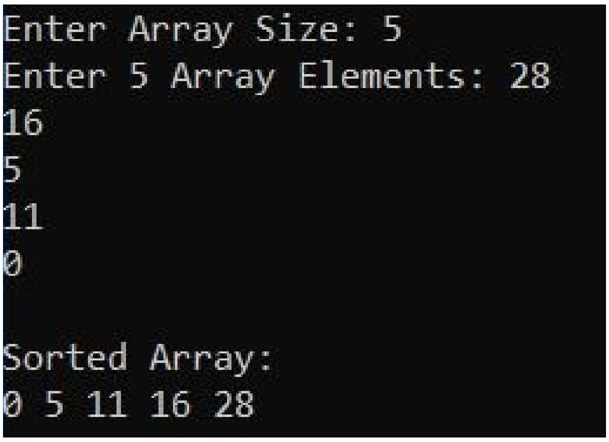
```
How many elements are u going to enter?: 9
Enter 9 elements: 15
34
75
48
67
18
34
37
42

Array before sorting:
15  34  75  48  67  18  34  37  42

Array after sorting:
15  18  34  34  37  42  48  67  75
```

Program 6: - Write a program in C++ to perform Insertion Sort.

Code:

```
#include<iostream>
using namespace std;
int main()
{
   int arr[50], tot, i, j, k, elem, index;
   cout<<"Enter the Size for Array: ";
   cin>>tot;
   cout<<"Enter "<<tot<<" Array Elements: ";
   for(i=0; i<tot; i++)
      cin>>arr[i];
   for(i=1; i<tot; i++)
   {
      elem = arr[i];
      if(elem<arr[i-1])
      {for(j=0; j<=i; j++)
         {if(elem<arr[j])
            {  index = j;
               for(k=i; k>j; k--)
                  arr[k] = arr[k-1];
               break;
            }}}
      else
         continue;
      arr[index] = elem;
   }
   cout<<"\nThe New Array (Sorted Array):\n";
   for(i=0; i<tot; i++)
      cout<<arr[i]<<"  ";
   cout<<endl;
   return 0;
}
```

Output:



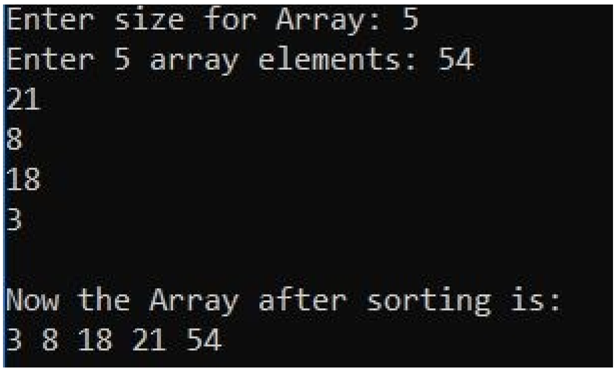
```
Enter Array Size: 5
Enter 5 Array Elements: 28
16
5
11
0

Sorted Array:
0 5 11 16 28
```

Program 7: - Write a program in C++ to perform Selection Sort.

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int tot, arr[50], i, j, temp, small, chk, index;
    cout<<"Enter the Size of Array: ";
    cin>>tot;
    cout<<"Enter "<<tot<<" Array Elements: ";
    for(i=0; i<tot; i++)
        cin>>arr[i];
    for(i=0; i<(tot-1); i++)
    {
        chk=0;
        small = arr[i];
        for(j=(i+1); j<tot; j++)
        {
            if(small>arr[j])
            {
                small = arr[j];
                chk++;
                index = j;
            }
        }
        if(chk!=0)
        {
            temp = arr[i];
            arr[i] = small;
            arr[index] = temp;
        }
    }
    cout<<"\nSorted Array is:\n";
    for(i=0; i<tot; i++)
        cout<<arr[i]<<" ";
    cout<<endl;
    return 0;
}
```

Output:



```
Enter size for Array: 5
Enter 5 array elements: 54
21
8
18
3

Now the Array after sorting is:
3 8 18 21 54
```