Practical No:- 1(A)

-

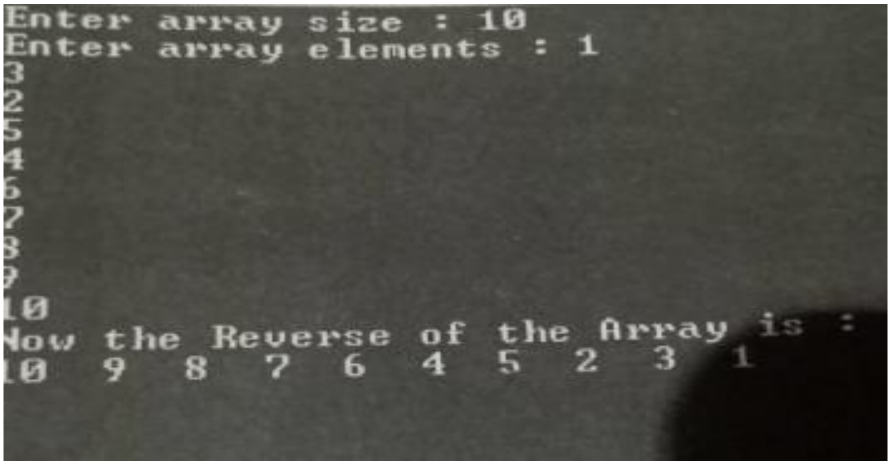
(a) Write a program to store the elements in 1-D array and perform the operations like searching, sorting, and reversing the elements.

Practical Implementation :-

Code:-

```
C++ Program - Reverse Array

#include<<iostream.h>
-
#include<<conio.h>

Void main()

{

Clrscr();

int arr[50], size, j, temp;

cout<<"Enter array size:";

cin>>size;

cout<<"Enter array elements:";

for(i=0; i<size; i++)

{

cin>>arr[i];

}
j=i-1; // now j will point to the last element
i=0; // and i will be point to the first element
while (i<j)
{
temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
i++;
j--;
}
cout<<"Now  the Reverse of the Array is":\n;
for(i=;i<size;i++)
{
cout<<arr[i]<<" ";
}
getch();
}
```

Output:-



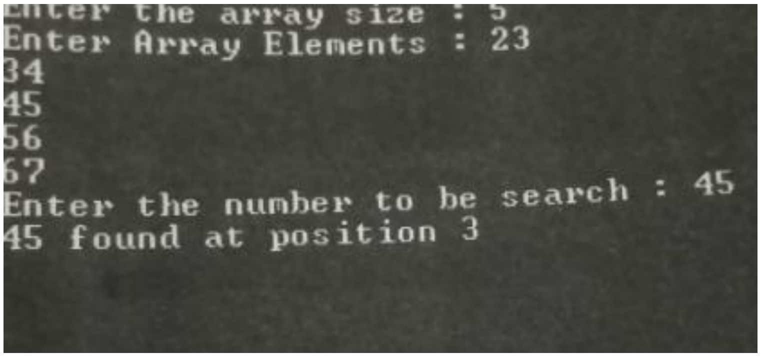
```
Enter array size : 10
Enter array elements : 1
3
2
5
4
6
7
8
9
10
Now the Reverse of the Array is :
10   9   8   7   6   4   5   2   3   1
```

C++ Program - Linear Search in Array

```
#include<iostream.h>

#include<conio.h>

void main()

{

clrscr();

int arr[10], i, num, n, c=0, pos;

cout<<"Enter the array size:";

cin>>n;

cout<<"Enter Arrray Elements:";

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

{

cin>>arr[i];

}

cout<<"Enter the number to be search:";

cin>>num;

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

{

if(arr[i]==num)

{

c=1;

pos=i+1;

break;
}

}

if(c==0)

{

cout<<"Number not found..!!"

}

else

{

cout<<num<<"found at position"<<pos;

}

getch();

}
```

Output:-

-

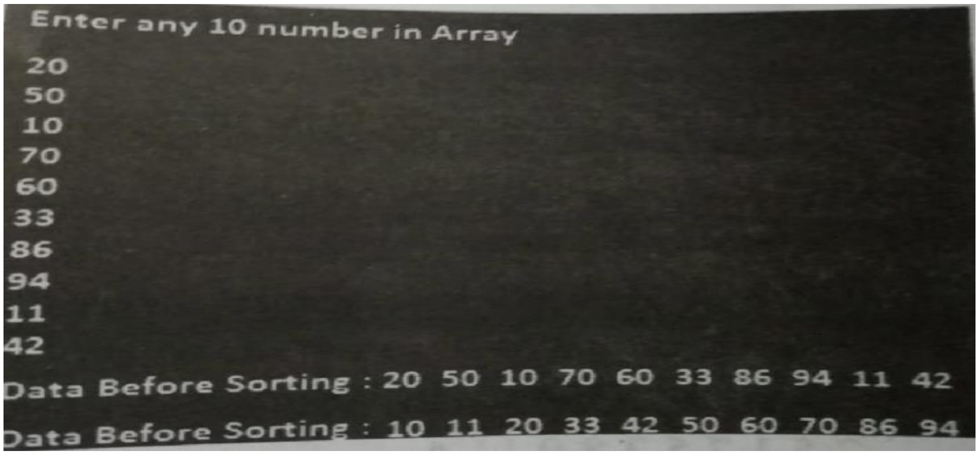


```
Enter the array size : 5
Enter Array Elements : 23
34
45
56
67
Enter the number to be search : 45
45 found at position 3
```

C++ Program - Sort Elements of Array in Ascending Order

```
#include<iostream.h>

#include<conio.h>

void main();

int i

a [10], temp, j;

clrscr();

cout<<"Enter any 10 num in array:\n ";

for(i=0; j<=10; i++)

{

cin>>a[i];

}

cout<<"\nData before sorting:";

for(j=0; j<10; j++)

{

cout<<a[j];

}

for(i=0;i<=10; i++)

{

for(j=0; j<=10 - i; j++)

{

if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1] = temp;
}
}
}
cout<<"\nData after sorting:";
for(j=0;j<10;j++)
{
cout<<a[j];
}
getch();
}
```

Output:-



```
Enter any 10 number in Array
20
50
10
70
60
33
86
94
11
42
Data Before Sorting : 20  50  10  70  60  33  86  94  11  42
Data Before Sorting : 10  11  20  33  42  50  60  70  86  94
```

Practical No:- 6(A)

-

(a) write a program to implement merge sort

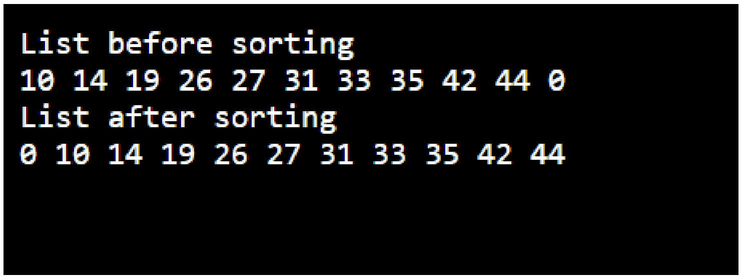
Practical Implementation :-

Code:-

```
#include <stdio.h>

#define max 10

int a[11] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0 };

int b[10];

void merging(int low, int mid, int high) {

  int l1, l2, i;

  for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) {

    if(a[l1] <= a[l2])

      b[i] = a[l1++];

    else

      b[i] = a[l2++];

  }

    while(l1 <= mid)

    b[i++] = a[l1++];

while(l2 <= high)

    b[i++] = a[l2++];

  for(i = low; i <= high; i++)

    a[i] = b[i];

}
```

```
void sort(int low, int high) {
   int mid;

   if(low < high) {
      mid = (low + high) / 2;
      sort(low, mid);
      sort(mid+1, high);
      merging(low, mid, high);
   } else {
      return;
   }
}
int main() {
   int i;
 printf("List before sorting\n");
    for(i = 0; i <= max; i++)
      printf("%d ", a[i]);
sort(0, max);
printf("\nList after sorting\n");
   for(i = 0; i <= max; i++)
      printf("%d ", a[i]);
}
```

Output:-



```
List before sorting
10 14 19 26 27 31 33 35 42 44 0
List after sorting
0 10 14 19 26 27 31 33 35 42 44
```

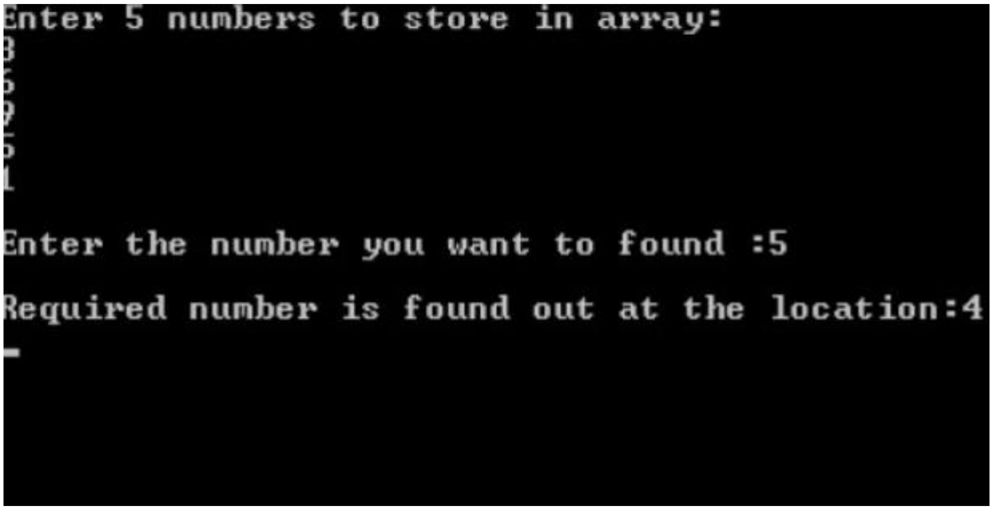
(B) write a program to search the element using sequential search

Practical Implementation :-

Code:-

```
#include<iostream.h>
#include<conio.h>
main()
{
int arr1[5];
int req;
int location=-5;
cout<<"Enter 5 numbers to store in array: "<<endl;
for(int i=0; i<5; i++)
{
cin>>arr1[i];
}
cout<<endl;
cout<<"Enter the number you want to find :";
cin>>req;
cout<<endl;

for(int w=0;w<5;w++)
{
if(arr1[w] == req)
location=w;
}
if(location !=-5)
{
cout<<"Required number is found out at the location:"<<location+1;
cout<<endl;
}
else
cout<<"Number is not found ";
getch();
}
```

Output:-



```
Enter 5 numbers to store in array:
3
6
9
5
1

Enter the number you want to found :5

Required number is found out at the location:4
_
```

(C) write a program to search the element using binary search

Practical Implementation :-

Code:-

Practical No:- 8(A)

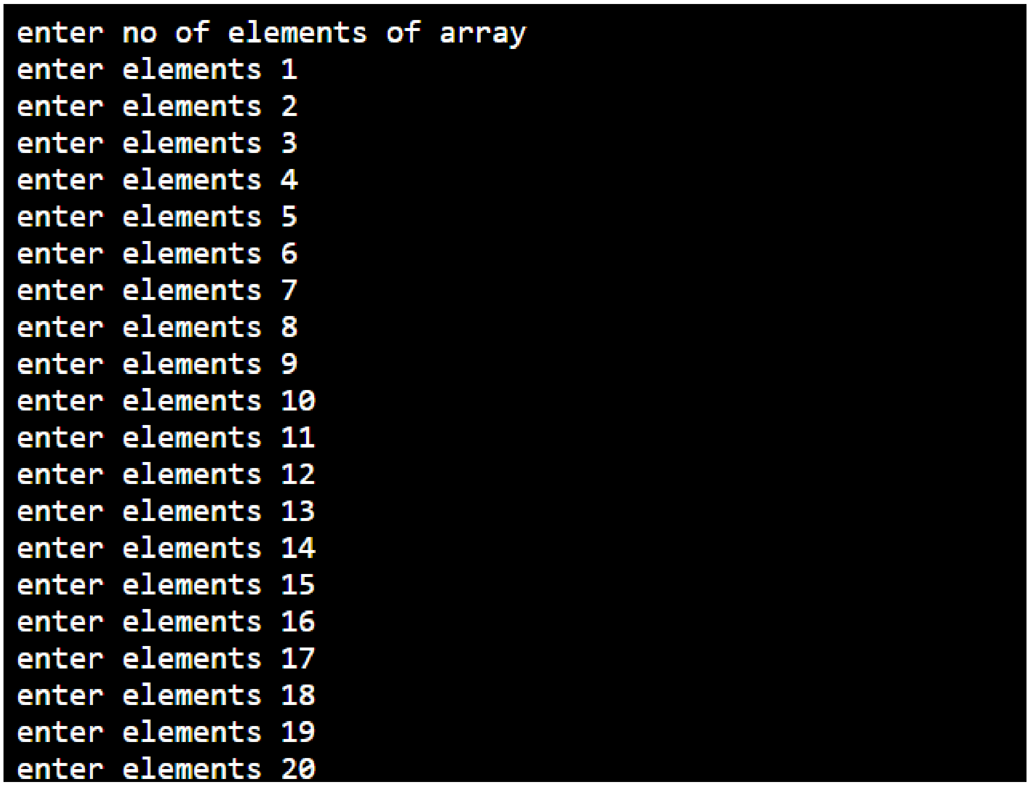
(A)write a program to insert the element into maximum heap

Practical Implementation :-

Code:-

```
#include <iostream>
using namespace std;
void max_heap(int *a, int m, int n) {
  int j, t;
  t = a[m];
  j = 2 * m;
  while (j <= n) {
    if (j < n && a[j+1] > a[j])
      j = j + 1;
    if (t > a[j])
      break;
    else if (t <= a[j]) {
      a[j / 2] = a[j];
      j = 2 * j;
    }
  }
  a[j/2] = t;
  return;
}
void build_maxheap(int *a,int n) {
  int k;
  for(k = n/2; k >= 1; k--) {
    max_heap(a,k,n);
  }
}
int main() {
  int n, i;
  cout<<"enter no of elements of array\n";
  cin>>n;
  int a[30];
  for (i = 1; i <= n; i++) {
    cout<<"enter elements"<<" "<<(i)<<endl;
    cin>>a[i];
  }
  build_maxheap(a,n);
  cout<<"Max Heap\n";
  for (i = 1; i <= n; i++) {
    cout<<a[i]<<endl;
  }
}
```

Output:-



```
enter no of elements of array
enter elements 1
enter elements 2
enter elements 3
enter elements 4
enter elements 5
enter elements 6
enter elements 7
enter elements 8
enter elements 9
enter elements 10
enter elements 11
enter elements 12
enter elements 13
enter elements 14
enter elements 15
enter elements 16
enter elements 17
enter elements 18
enter elements 19
enter elements 20
```

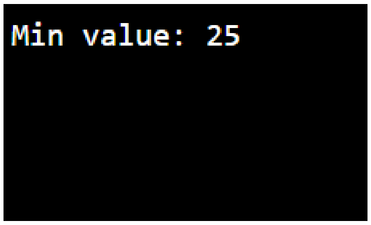
(B)write a program to insert the element into minimum heap.

Practical Implementation :-

Code:-

```
#include <iostream>
#define SIZE(arr) (sizeof(arr) / sizeof(arr[0]))
using namespace std;
int getMinElement(int *heap, int n){
  int minElement = heap[n / 2];
  for (int i = n / 2 + 1; i < n; ++i) {
    minElement = min(minElement, heap[i]);
  }
  return minElement;
}
int main(){
  int heap[] = {120, 90, 100, 70, 75, 80, 60, 25, 40, 35};
  cout << "Min value: " << getMinElement(heap, SIZE(heap)) <<
"\n";
  return 0;
}
```

Output:-



```
Min value: 25
```