Implementation of searching and sorting algorithms

1. Implementation of Linear Search

Aim :

Write a C program to implement the linear search

Algorithm:

```
1. Declare the input variables
2. Input the data into array
3. Input key for search the data in the array
4. Start comparing the key value with each data in the array as follows
   for(i=0;i<n;i++) if(a[i]==x) { print( Data available);find++;}
   if(find==0)
   print("Data Unavailable");
5. Display the result.
```

Program :

```
#include<stdio.h>
#include<conio.h>
int a[20],find=0,x;
main()
{
int i,n;
clrscr();
printf("\nEnter the size\n");
scanf("%d",&n);
printf("\nEnter the elements one by one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("\nEnter the element to be searched :");
scanf("%d",&x);
for(i=0;i<n;i++)
if(a[i]==x)
{
printf("\nThe element %d is at %d ",x,i+1);
find++;
}
if(find==0)
printf("\nThe given element %d is not in the array",x);
getch();
}
```

Sample Input and Output:

```
Enter the size
 5
 Enter the elements one by one
 11
 22
 33
```

```
44
55
Enter  the element to be searched
33
The element 33 is at 3
```

Result :

Thus, the Implementation of linear search by C programming is done successfully.

2. Implementation of Binary Search

Aim :

To Write a C program to implement the binary search.

Algorithm :

```
1. Declare the input variables
2. Input the data into array
3. Input key for search the data in the array
4. Start comparing the key value with each data in the array as follows
   low=0;  high=n-1;
   while(low<=high)
   mid=(low+high)/2;
   if(x>a[mid]) low=mid+1;
   else if(x<a[mid]) high=mid-1;
   else { print( Date available); find++; break;
   if(find==0)
   print(Data Unavailable);
5. End the Program
```

Program :

```
#include<stdio.h>
#include<conio.h>
int a[20],find=0,x;
main()
{
int i,n,low,high,mid;
clrscr();
printf("\nEnter the size\n");
scanf("%d",&n);
printf("\nEnter the elements one by one\n");
for(i=0;i<n;i++)
{
printf("\na[%d]=",i);
scanf("%d",&a[i]);
}
printf("\nEnter the element to be searched :");
scanf("%d",&x);
low=0;
high=n-1;
while(low<=high)
{
mid=(low+high)/2;
if(x>a[mid])
low=mid+1;
else if(x<a[mid])
high=mid-1;
else
{
printf("\nThe element %d is at position %d",x,mid);
find++;
break;
}
}
if(find==0)
printf("\nThe given element %d is not in the array",x);
getch();
}
```

Sample Input and Output:

```
   Enter the size
   5
   Enter the elements one by one
   11
   22
   33
   44
   55
    Enter  the element to be searched
    33
    The element 33 is at position 3
```

Result:

Thus, the Implementation of binary search by C programming is done successfully.

3. Insertion Sort

Aim:

To Write a C program to implement the insertion sort

Algorithm:

1. Read the number of elements you need (n).
2. Read the elements one by one by using for loop.
3. Start comparing the second element to all front element.
4. Increment the front element.
5. Print the sorted elements which are must in ascending order.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,k,x,t,n,a[30];
clrscr();
printf("\nEnter the Number Of elements ");
scanf("%d",&n);
printf("Enter the elements one by one ");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(k=1;k<n;k++)
{
t=a[k];
for(i=k-1;i>=0;i--)
if(a[i]>t)
{
a[i+1]=a[i];
a[i]=t;
}
}
printf("\nThe sorted element \n");
for(i=0;i<=n;i++)
{
printf("\t%d",a[i]);
}
getch();
}
```

Sample Input and Output:

```
Enter the number of elements to be sorted:6
Enter the elements:4
5
7
3
8
6
Sorted array by using bubble sort is : 3     4       5       6       7       8
```

Result :

Thus, Insertion sort method through C programming is done successfully.

4. Selection Sort

Aim:

Write a C program to implement the selection sort

Algorithm:

1.Read the number of elements.
2.Read the elements one by one using for loop.
3.Sort the elements in ascending order using nested for loops.
4.Display the result.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,n,t,u,a[20];
int selectposition;
int selectelement;
clrscr();
printf("\nEnter the Number Of Elements: ");
scanf("%d",&n);
printf("\nEnter the elements one by one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=n-1;i>0;i--)
{
selectposition=0;
selectelement=a[0];
for(j=1;j<=i;j++)
{
if(a[j]>selectelement)
{
selectelement=a[j];
selectposition=j;
}
}
a[selectposition]=a[i];
a[i]=selectelement;
}printf("\n  THE SORTED ARRAY\n");
for(i=0;i<n;i++)
printf("\t%d",a[i]);
getch();
}
```

Sample Input and Output:

```
Enter the number of elements to be sorted:6
Enter the elements:4
5
7
3
8
```

```
6
Sorted array by using bubble sort is : 3     4       5       6       7       8
```

Result :

Thus, selection sort method through C programming is done successfully

5. Bubble Sort

Aim:

To Write a C program to implement the bubble sort

Algorithm:

1. Declare the input variables
2. Input the data into array
3. Start comparing the data with adjacent data in the array as follows
   ```
   for(i=n-1;i>0;i--)
   {
   for(j=0;j<i;j++) if(a[j]>a[j+1])
   {
   temp=a[j];  a[j]=a[j+1];  a[j+1]=temp;
   }
   ```
4. Display the result.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,a[20],temp,n;
clrscr();
printf("\n Enter the Nuber of elements: ");scanf("%d",&n);
printf("\nEnter the elements one bye one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=n-1;i>0;i--)
for(j=0;j<i;j++)
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
printf("\n  THE SORTED ARRAY \n");
for(i=0;i<n;i++)
{  printf("\t%d",a[i]); }
getch();
}
```

Sample Input and Output:

```
Enter the number of elements to be sorted:6
```

```
Enter the elements:4
5
7
3
8
6
Sorted array by using bubble sort is : 3    4      5      6      7      8
```

Result :

Thus, bubble sort method through C programming is done successfully