Page No.
Date

$\frac{11}{11}\frac{8}{9}$

Experiment No 10

Aim :- Implementation of selection sorting technique considering a real world applicati

Objective :- To impart knowledge of sorting and searching algo.

Theory :-

Introduction to sorting :- Sorting refers to arranging data in a particular format sorting algorithm specifies the way to arrange data in a particular order. Importance of sorting lies in the fact that data searching can be optimized to very high level in more readable formats.

Types of sorting :-

1. Bubble sort
2. Selection sort
3. Insertion sort
4. Merge sort

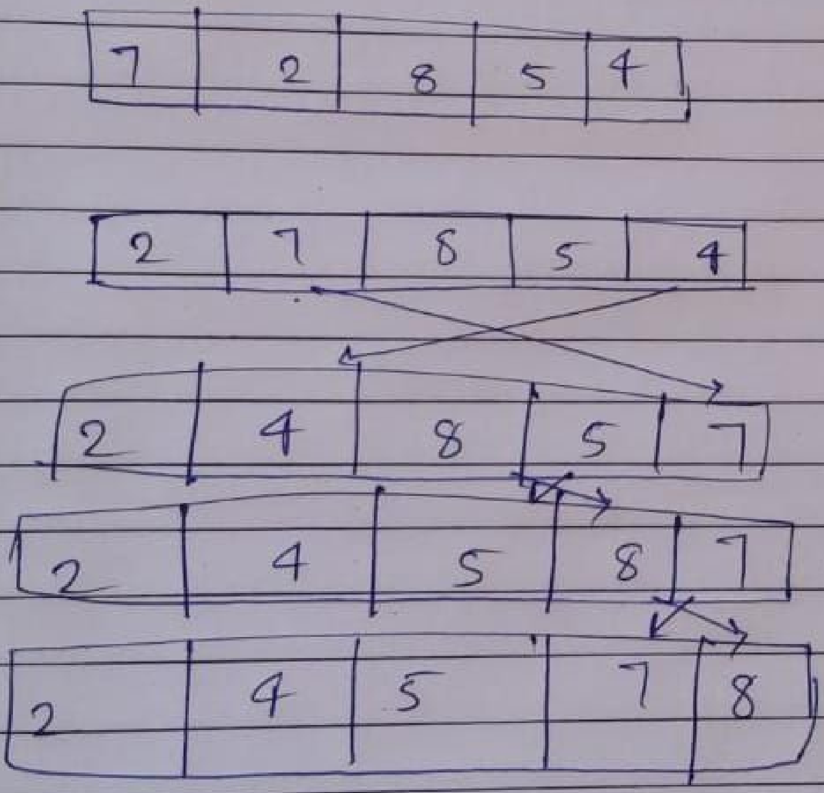
Introduction of selection sort

In selection sort, the first smallest element is selected from the unsorted array and placed at first position. After that second smallest element is selected and placed in second position.

process continues until the arrays is entirely sorted. Average and worst-case complexity of selection sort is $O(n^2)$.

```
Algorithm- Selection Sort (A[0....n-1])
  for i ← 0 to n-2 do
        min ← 1
     for j ← i+1 to n-1 do
    if A[j] < A[min] min ← j
    swap A[i] and A[min]
```

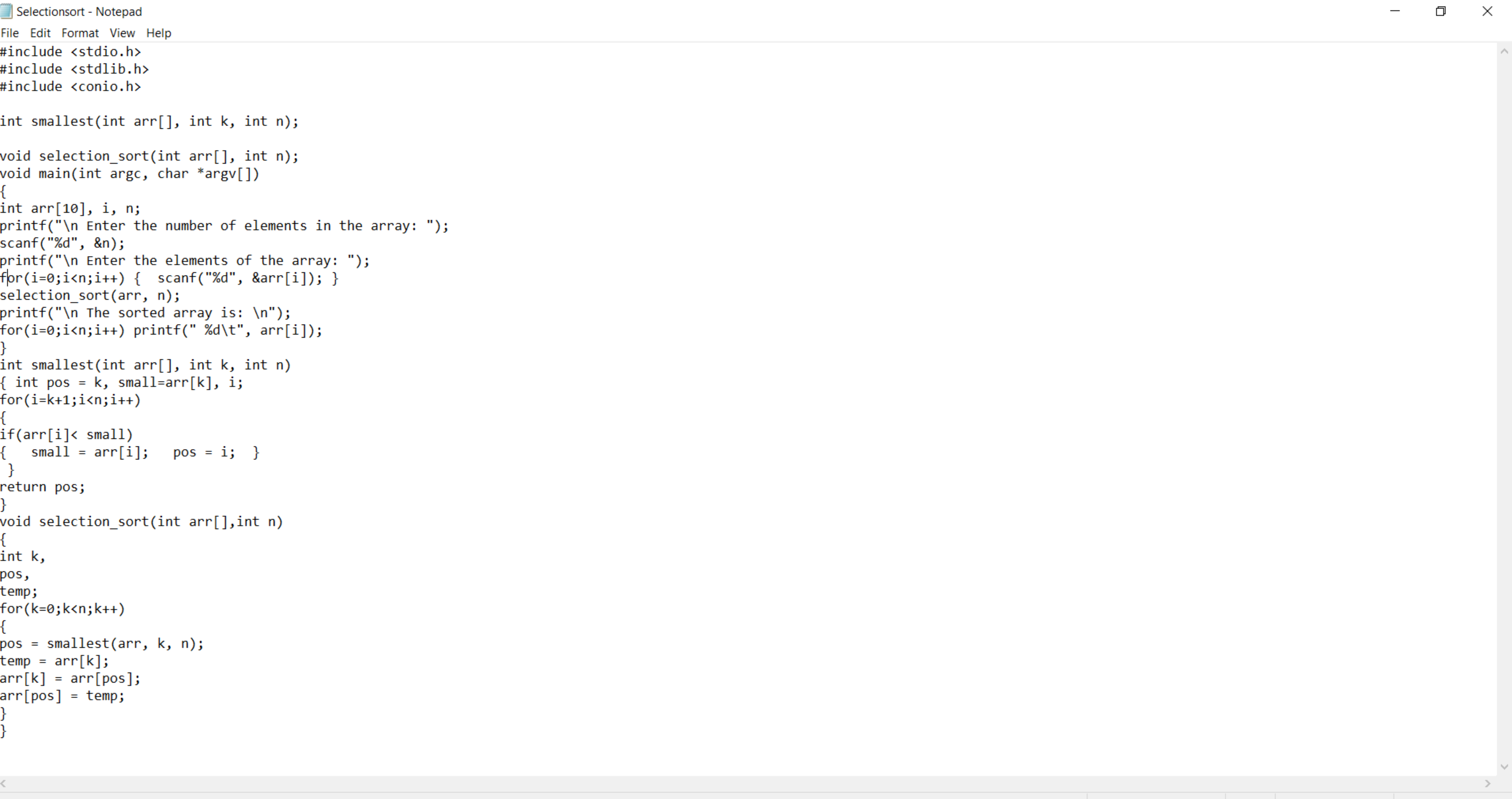
Example:-



final sorted array

Conclusion:- Studied how to impart knowledge of sorting and searching algorithms through implementation of selection sorting techniques.

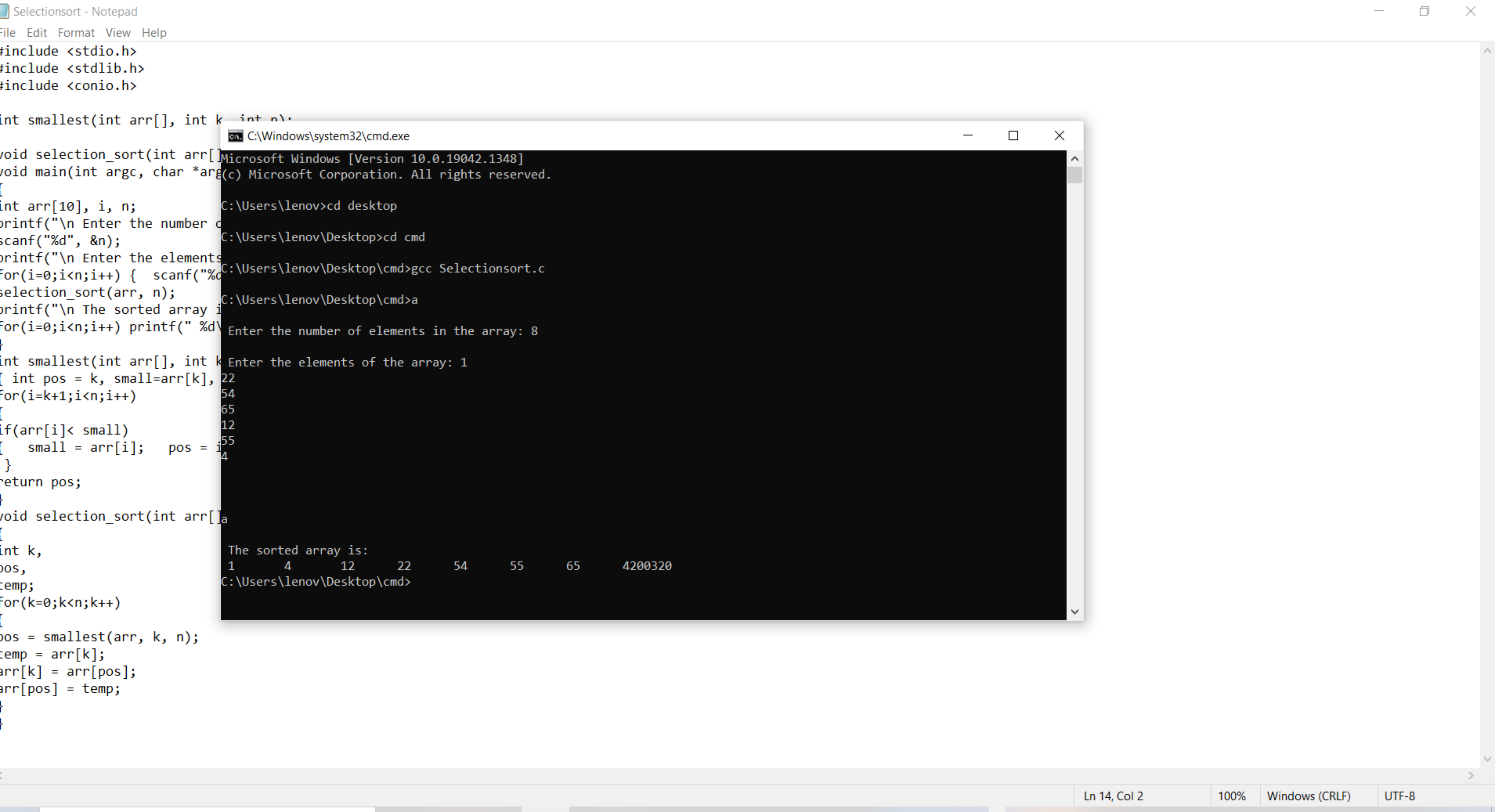
Outcome: Implement sorting and searching technique for real-world applications



```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int smallest(int arr[], int k, int n);

void selection_sort(int arr[], int n);
void main(int argc, char *argv[])
{
int arr[10], i, n;
printf("\n Enter the number of elements in the array: ");
scanf("%d", &n);
printf("\n Enter the elements of the array: ");
for(i=0;i<n;i++) {  scanf("%d", &arr[i]); }
selection_sort(arr, n);
printf("\n The sorted array is: \n");
for(i=0;i<n;i++) printf(" %d\t", arr[i]);
}
int smallest(int arr[], int k, int n)
{ int pos = k, small=arr[k], i;
for(i=k+1;i<n;i++)
{
if(arr[i]< small)
{   small = arr[i];   pos = i;  }
 }
return pos;
}
void selection_sort(int arr[],int n)
{
int k,
pos,
temp;
for(k=0;k<n;k++)
{
pos = smallest(arr, k, n);
temp = arr[k];
arr[k] = arr[pos];
arr[pos] = temp;
}
}
```



```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int smallest(int arr[], int k, int n);

void selection_sort(int arr[
void main(int argc, char *arg
{
int arr[10], i, n;
printf("\n Enter the number o
scanf("%d", &n);
printf("\n Enter the elements
for(i=0;i<n;i++) {  scanf("%
selection_sort(arr, n);
printf("\n The sorted array i
for(i=0;i<n;i++) printf(" %d\
}
int smallest(int arr[], int k
{ int pos = k, small=arr[k],
for(i=k+1;i<n;i++)
{
if(arr[i]< small)
{   small = arr[i];   pos = i
 }
return pos;
}
void selection_sort(int arr[]
{
int k,
pos,
temp;
for(k=0;k<n;k++)
{
pos = smallest(arr, k, n);
temp = arr[k];
arr[k] = arr[pos];
arr[pos] = temp;
}
}
```

```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenov>cd desktop

C:\Users\lenov\Desktop>cd cmd

C:\Users\lenov\Desktop\cmd>gcc Selectionsort.c

C:\Users\lenov\Desktop\cmd>a

 Enter the number of elements in the array: 8

 Enter the elements of the array: 1
22
54
65
12
55
4




a

 The sorted array is:
 1       4       12      22      54      55      65      4200320
C:\Users\lenov\Desktop\cmd>
```