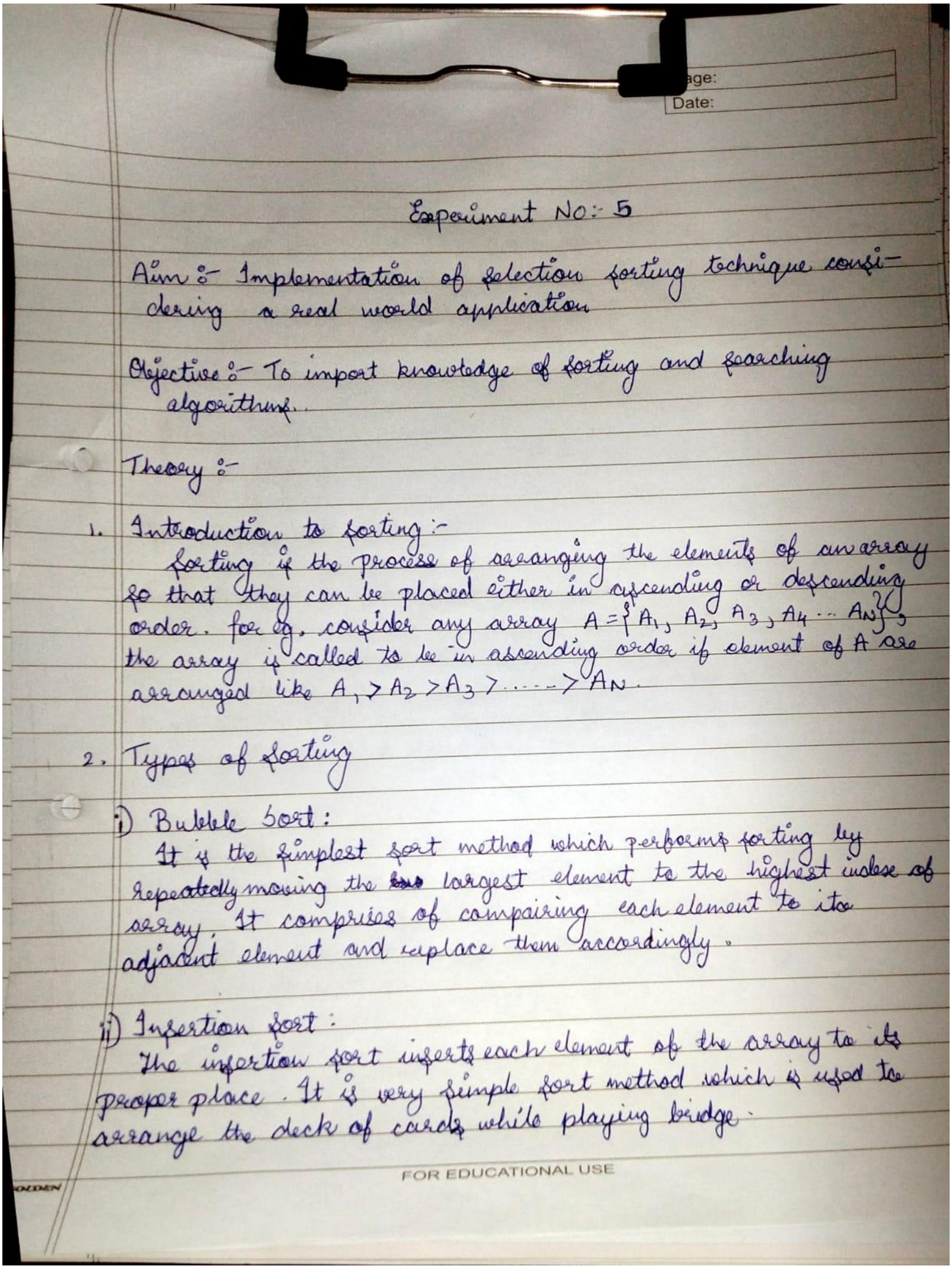


Experiment No:- 5

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

Objective :- To import knowledge of sorting and searching algorithms.

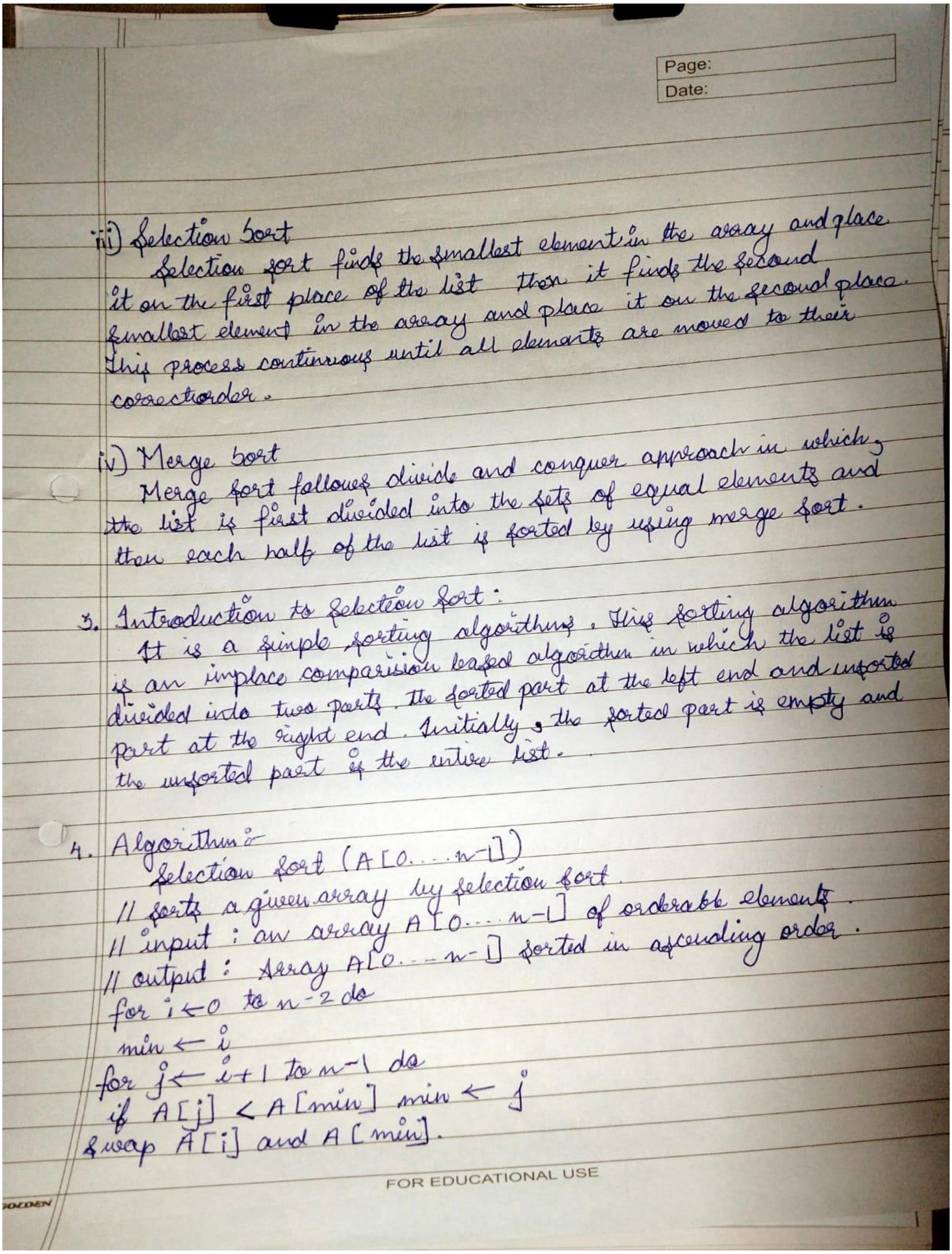
Theory :-

1. Introduction to sorting:-
sorting is the process of arranging the elements of an array so that they can be placed either in ascending or descending order. for eg. consider any array $A = \{A_1, A_2, A_3, A_4 \cdots A_N\}$, the array is called to be in ascending order if element of A are arranged like $A_1 > A_2 > A_3 > \ldots > A_N$.

2. Types of sorting

i) Bubble Sort:
It is the simplest sort method which performs sorting by repeatedly moving the ~~[illegible]~~ largest element to the highest index of array. It comprises of compairing each element to its adjacent element and replace them accordingly.

ii) Insertion sort:
The insertion sort inserts each element of the array to its proper place. It is very simple sort method which is used to arrange the deck of cards while playing bridge.



Page:
Date:

iii) Selection sort

Selection sort finds the smallest element in the array and place it on the first place of the list then it finds the second smallest element in the array and place it on the second place. This process continuous until all elements are moved to their correctorder.

iv) Merge sort

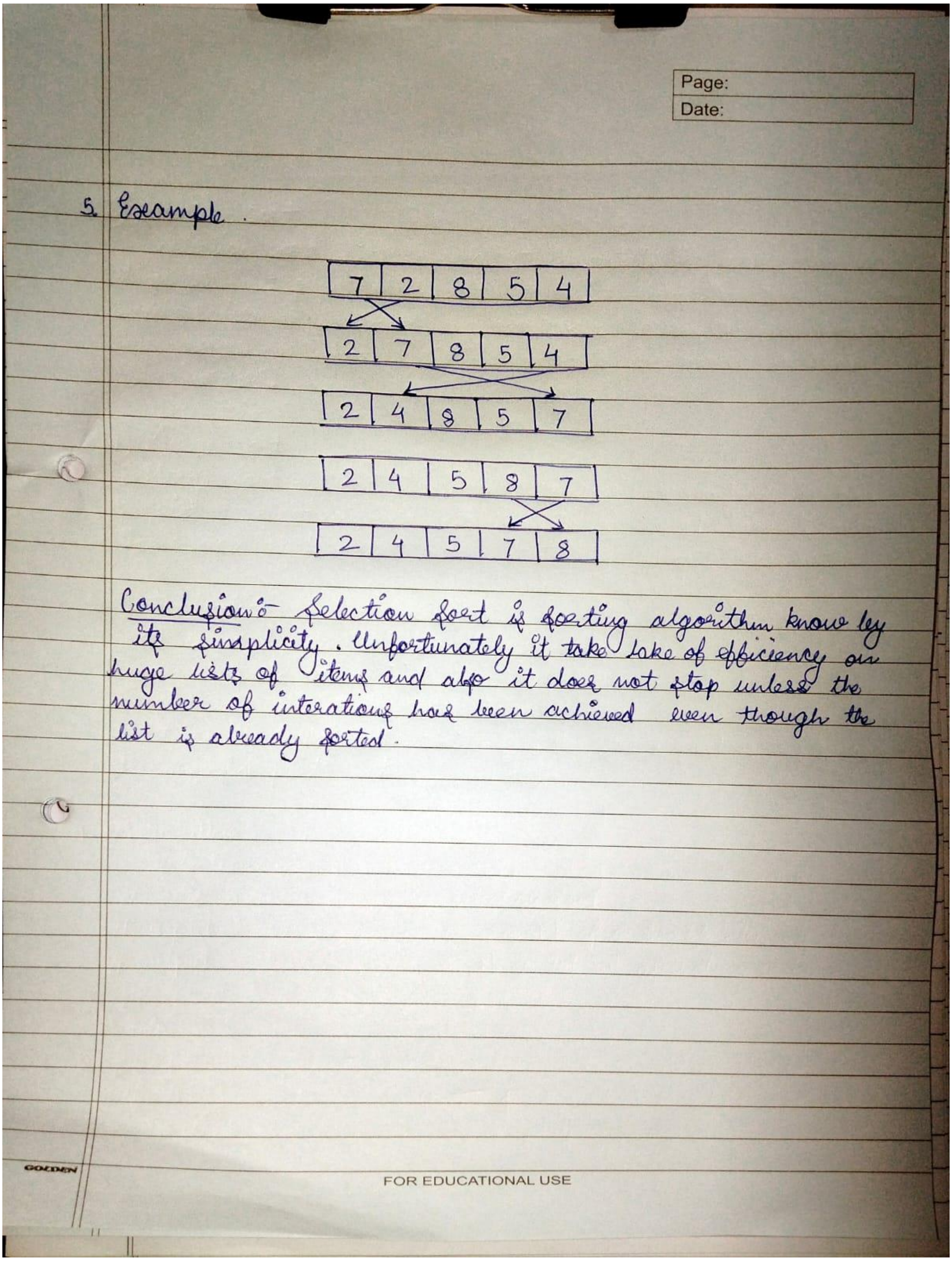
Merge sort follows divide and conquer approach in which, the list is first divided into the sets of equal elements and then each half of the list is sorted by using merge sort.

3. Introduction to selection sort:

It is a simple sorting algorithms. This sorting algorithm is an inplace comparision based algorithm in which the list is divided into two parts. the sorted part at the left end and unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list.

4. Algorithm:-

```
Selection sort (A[0....n-1])
// sorts a given array by selection sort.
// input : an array A[0....n-1] of orderable elements.
// output : Array A[0....n-1] sorted in ascending order.
for i←0 to n-2 do
    min ← i
for j← i+1 to n-1 do
    if A[j] < A[min] min ← j
swap A[i] and A[min].
```



Page:

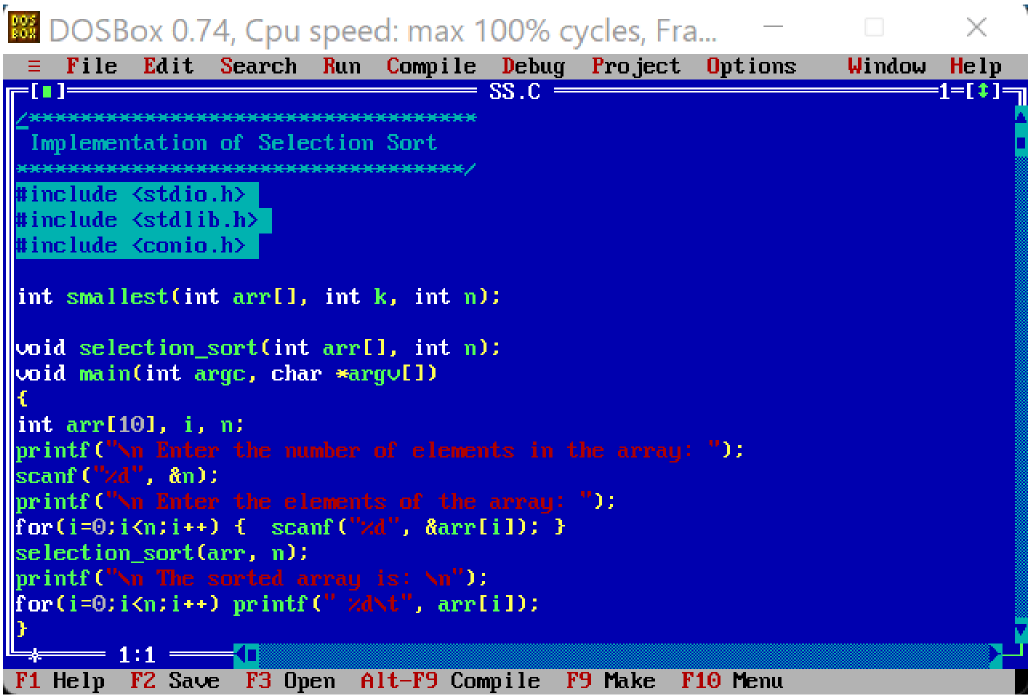
Date:

5. Example.

7	2	8	5	4
2	7	8	5	4
2	4	8	5	7
2	4	5	8	7
2	4	5	7	8

Conclusion:- Selection Sort is Sorting algorithm know by its Simplicity. Unfortunately it take lake of efficiency on huge lists of items and also it does not stop unless the number of interations has been achieved even though the list is already sorted.

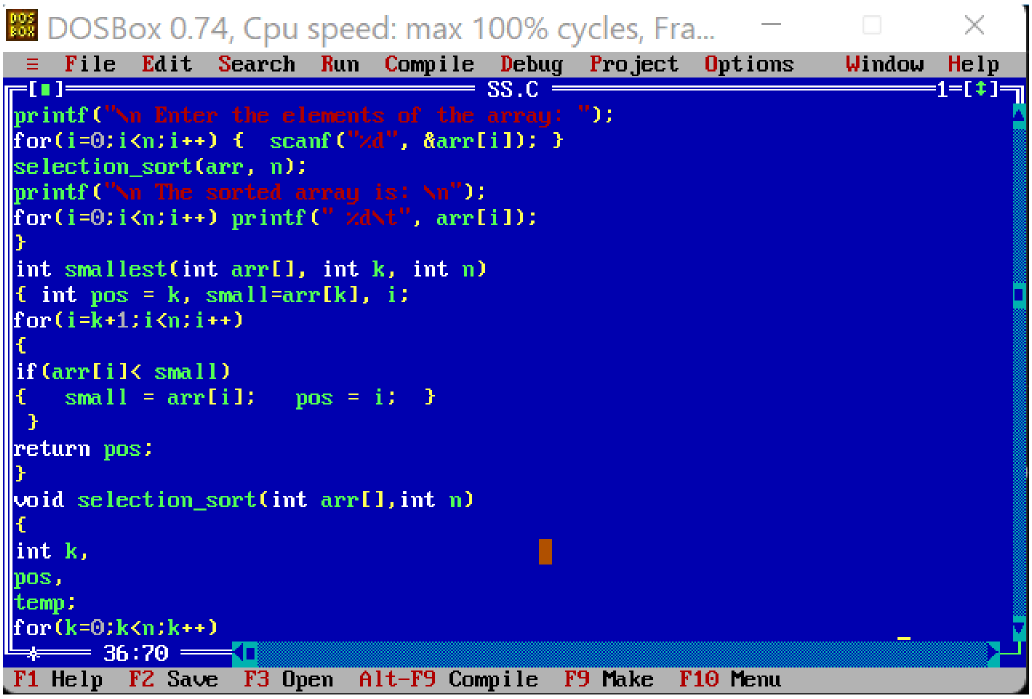
***Implementing selection sort**


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SS.C 1=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

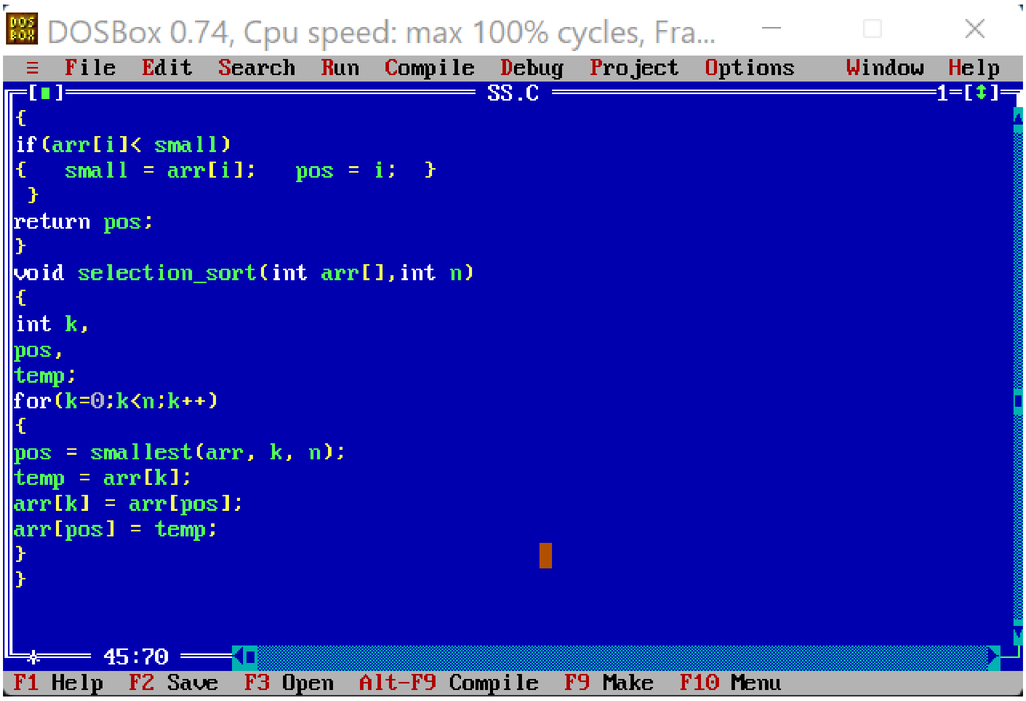
```
/************************************
 Implementation of Selection Sort
************************************/
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

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]);
}
```

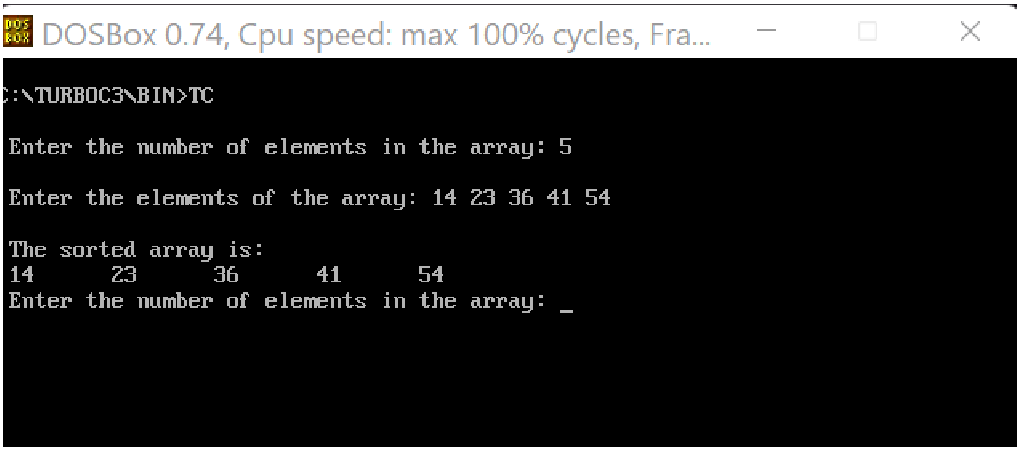

DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SS.C 1=[↕]
36:70
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
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++)
```


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SS.C 1=[↕]
45:70
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
{
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;
}
}
```

*Output :-



```
:\TURBOC3\BIN>TC

Enter the number of elements in the array: 5

Enter the elements of the array: 14 23 36 41 54

The sorted array is:
14      23      36      41      54
Enter the number of elements in the array: _
```