Aim: Write a menu driven program to sort an Array using Selection sort.

Objective: To implement sorting algorithm having time complexity of $O(n^2)$ in all cases.

Aim : Write a menu driven program to sort an array using Selection Sort.

Objective: To implement sorting Algorithm having time Complexity of $O(n^2)$ in all cases.

Theory: Selection sort is a straight forward & intutive Algorithm that operates by dividing the input list into two parts i.e a sorted & a unsorted section. The Algorithm iteratively selects the smallest elements from unsorted portion & swaps it with the first element of section.
This process continues until the entire list is sorted. The key is to gradually build up a sorted sequence by repeatedly identifying the smallest element from the unsorted part & placing it at the begining of the sorted portion.

Selection sort is an in-place sorting algorithm, meaning it sorts the elements within the existing data structure without requiring additional memory.

Program Code :

```
#include <stdio.h>
void selectionSort (int arr [] , int n) {
int i, j , minIndex , temp;

for (i = 0; i < n - 1 ; i++) {
    minIndex = i;
        for( j = i+1 ; j<n ; j++) {
            if (arr[j] < arr [minIndex]) {
                minIndex = j;
            }
        }

    temp = arr [i] ;
    arr [i] = arr [minIndex];
    arr [minIndex ] = temp;
  }
}

void printArray (int arr[] , int size) {
    for (int i=0; i<size ; i++ )
        {
            printf ("%d", arr [i]);
        }

    printf ("\n");
}
```

```
~~void~~
int main () {
   int n;

   printf ("Enter the number of elements \n");
   scanf ("%d", &n);

   int arr [n];

   printf ("Original Array is :\n");
   printfArray (arr, n);

   selectionSort (arr, n);

   printf ("Sorted array is:\n");
   printArray (arr, n);

   return 0;
}
```

Output:

```
Enter the number of elements : 5
Enter the elements :
Element 1 : 50
Element 2 : 40
Element 3 : 30
Element 4 : 20
Element 5 : 10
Original Array : 50 40 30 20 10
Sorted Array : 10 20 30 40 50
```

Algorithm:- i] Input

Take an array of elements.

ii] Outer loop

Iterate through the array from the first element to second element till the last elements.

iii] Assume Minimum

Assume the current index (i) the index of minimum element.

iv] Inner Loops

Start an inner loops from the next element to last element. Compare the element at index 'j' with element at the assumed minimum index

v] Swap

After the inner loop, if the assumed minimum

Conclusion : Program of selection sort is implemented & executed successfully using C Programming Langauge.

is minimum from the current index, also swap the elements at the two indices.

vi] Repeat

Repeat steps 3-5 for each iteration of the outer loop.

vii] Output

The array is sorted when outer loop completes.

Conclusion: Program of selection sort is implemented & executed sucessfully using C Programming Language.