Subject :			Software :	
			Hardware :	
Branch :	Semester :		Page No.	Prog No.

PROBLEM STATEMENT Sort a given set of an Integer elements using selection sort method in C language and compute its time Complexity. Run the program for varied values of n> 5000 and record the time taken to Sort. plot a graph of the time taken Vs n. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the brute force method works along with its time complexity analysis: Worst case, average case and best case.

ALGORITHM & CODE :

```
#include <stdio.h>
# include <stdlib.h>
# include < time.h>
void selection sort (int arr[], intn)
{ int i, j, min_idx;
   for (i=0; i<n-1 ; i++)
   { min_idx =i;
     for (j=i+1 ; j<n; j++)
     { if (arr[j] < arr[min_idx])
         { min_idx =j;
         } ~~for (j=i+1; j<n; j++)~~
     }
     int temp = arr[min_idx];
     arr[min_idx] = arr[i];
     arr[i] = temp;
   }
 }
```

INPUT GIVEN	
OUTPUT OBTAINED	
REMARKS	
GRADE :	Signature of Faculty Date : / Signature of Student Date :

Subject :		Software :	
		Hardware :	
Branch :	Semester :	Page No.	Prog No.

ATEMENT

S CODE :

```
void printArray(int arr[], int size)
{ int i;
  for (i=0; i<10; i++)
  {   printf("%d", arr[i]);
   }
   printf("\n");
}
int main()
{ int n;
   printf("Enter the number of elements in the
                array (>=5000):");
   scanf("%d", &n);
  if (n<5000)
 {  printf("please enter a number greater than
           or equal to 5000.\n");
     return 1;
  }
 int *arr = (int *)malloc(n*sizeof(int));
 srand(time(0));
 for(int i=0; i<n; i++)
 {
```

EN

TAINED

Signature of Faculty

Signature of Student

	Subject :		Software :	
			Hardware :	
	Branch :	Semester :	Page No.	Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
for(int i=0; i<n; i++)
{  arr[i] = rand() % 100000;
}
printf("Original array (first 10 elements):\n");
printArray(arr, n);
clock_t start = clock();
selectionSort(arr, n);
clock_t end = clock();

double time_taken = ((double)(end-start))/
                        CLOCKS_PER_SEC;

Printf("Sorted array (first 10 elements):\n");
PrintArray(arr, n);
Printf("Time taken by selection Sort : %f seconds\n",
                        time_taken);

free(arr);
return 0;
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

Signature of Faculty

Date :

Signature of Student

Date :

GRADE :

Subject :		Software :	
		Hardware :	
Branch :	Semester :	Page No.	Prog No.
PROBLEM STATEMENT			

ALGORITHM & CODE :

Output obtained

```
enter the no.of elements in the array (>=5000)=
                                        5000
original array (first 10 elements):
88513  56333  42454  48955  24357  12817
7743  58778  88879  90390
Sorted array :
5 47 71 82 122 123 253 284 302 306
time taken by selection sort : 0.037981 seconds.

enter the no.of elements in the array (>=5000) =
                                        10000
original array (first 10 elements):
46671  66401  2557  93704  82723  68949
46672  63104  20956  35347
Sorted array :
12 27 63 76 79 94 104 117 136 140
time taken by selection sort : 0.150059 seconds.
```

INPUT GIVEN		
OUTPUT OBTAINED		
REMARKS		
GRADE :	Signature of Faculty Date :	Signature of Student Date :

	Subject :		Software :	
			Hardware :	
	Branch :	Semester :	Page No.	Prog No.
PROBLEM STATEMENT				

ALGORITHM & CODE :

```
enter the no.of elements in the array (>=5000):
                                          15000
original array (first 10 elements):
45638  32426  95468  78663  61125 22864
9129   21724  34483  49603  .
Sorted array:
0  3  13  16  17  34  46  53  75  77
time taken by selection sort: 0.336844 seconds.

enter the no.of elements in the array (>=5000): 20000
original array (first 10 elements):
29179  44932  57836  4160  26248  64845
71710  67647  37621  72906
Sorted array:
0 1 5 9 22 24 27 28 29 29
time taken by selection sort: 0.597810 seconds.

enter the no.of element in the array (>=5000):
                                          25000
original array (first 10 elements):
89035  29639  56248  81693  41791  21034
```

INPUT GIVEN	
OUTPUT OBTAINED	
REMARKS	
GRADE :	Signature of Faculty Date :
	Signature of Student Date :

	Subject :	Software :
		Hardware :
Branch :	Semester :	Page No. / Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
852524   46343   81843   98761
Sorted array:
6 6 7 8 11 18 26 31 33 34
time taken by selection sort : 0.933049 seconds.
```

Time Complexity Analysis

Selection Sort time complexity for different cases:

- Best Case (Already Sorted) : $O(n^2)$
- Worst case (Reverse sorted): $O(n^2)$
- Average Case (Random Data): $O(n^2)$

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty
Date :

Signature of Student
Date :