

## LAB 05:

AIM: Sort a given set of N integer elements using Selection Sort technique and compute its time taken.

**ALGORITHM** : sel\_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

    small\_pos=i

    for j=i+1 to n-1 do

        if a[j]<a[small\_pos]

            small\_pos=j

        end if

    end for

    swap a[i] and a[small\_pos]

end for

**Program:**

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

```
#include<time.h>
```

```
#include<stdlib.h> /* To recognise exit function when compiling with gcc*/
```

```
void selsort(int n,int a[]);
```

```
void main()
```

```
{
```

```
int a[15000],n,i,j,ch,temp;
```

```
clock_t start,end;
```

```
while(1)
```

```
{
```

```

printf(&quot;\n1:For manual entry of N value and array elements&quot;);

printf(&quot;\n2:To display time taken for sorting number of elements N in the range 500 to
14500&quot;);

printf(&quot;\n3:To exit&quot;);

printf(&quot;\nEnter your choice:&quot;);

scanf(&quot;%d&quot;, &amp;ch);

switch(ch)
{
case 1: printf(&quot;\nEnter the number of elements: &quot;);

scanf(&quot;%d&quot;,&amp;n);

printf(&quot;\nEnter array elements: &quot;);

for(i=0;i<n;i++)
{
scanf(&quot;%d&quot;,&amp;a[i]);
}

start=clock();

selsort(n,a);

end=clock();

printf(&quot;\nSorted array is: &quot;);

for(i=0;i<n;i++)

printf(&quot;%d\t&quot;,a[i]);

printf(&quot;\n Time taken to sort %d numbers is %f Secs&quot;,n, (((double)(end-
start))/CLOCKS_PER_SEC));

break;

```

```

case 2:
n=500;
while(n<=14500) {
for(i=0;i<n;i++)
{
//a[i]=random(1000);
a[i]=n-i;
}
start=clock();
selsort(n,a);
//Dummy loop to create delay
for(j=0;j<500000;j++){ temp=38/600;}
end=clock();

printf("&quot;\n Time taken to sort %d numbers is %f Secs&quot;;n, (((double)(end-
start))/CLOCKS_PER_SEC));

n=n+1000;
}
break;
case 3: exit(0);
}
getchar();
}
}

void selsort(int n,int a[])
{

```

```
int i,j,t,small,pos;
for(i=0;i<n-1;i++)
{
    pos=i;
    small=a[i];
    for(j=i+1;j<n;j++)
    {
        if(a[j]<small)
        {
            small=a[j];
            pos=j;
        }
    }
    t=a[i];
    a[i]=a[pos];
    a[pos]=t;
}
```

## OUTPUT:

```
1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
Enter your choice: 1

Enter the number of elements: 4

Enter array elements: 44 33 22 11

Sorted array is: 11      22      33      44
Time taken to sort 4 numbers is 0.000000 Secs

1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
Enter your choice: 2

Time taken to sort 500 numbers is 0.000000 Secs
Time taken to sort 1500 numbers is 0.001000 Secs
Time taken to sort 2500 numbers is 0.002000 Secs
Time taken to sort 3500 numbers is 0.004000 Secs
Time taken to sort 4500 numbers is 0.007000 Secs
Time taken to sort 5500 numbers is 0.010000 Secs
Time taken to sort 6500 numbers is 0.015000 Secs
Time taken to sort 7500 numbers is 0.020000 Secs
Time taken to sort 8500 numbers is 0.024000 Secs
Time taken to sort 9500 numbers is 0.030000 Secs
Time taken to sort 10500 numbers is 0.037000 Secs
Time taken to sort 11500 numbers is 0.045000 Secs
Time taken to sort 12500 numbers is 0.053000 Secs
Time taken to sort 13500 numbers is 0.061000 Secs
Time taken to sort 14500 numbers is 0.071000 Secs

1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
```

# GRAPH:

