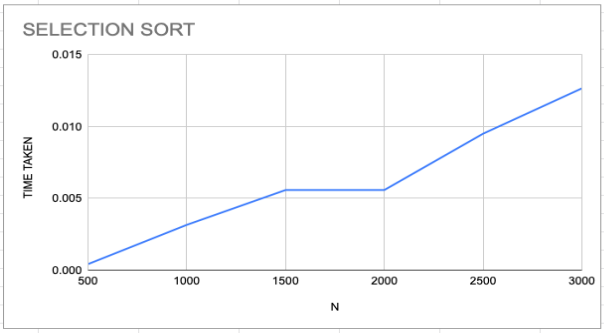
LAKSHMI S KUMAR
1BM19CS078
ADA LAB-3

```
        #include<stdio.h>

#include <stdlib.h>
#include <time.h>
clock_t start, end;
double cpu_time;
int main()
{
   int n,i,j,pos,temp,small,arr[10000],c,d;
   srand(time(0));
   printf("Enter the number of elements in array \n");
   scanf("%d", &n);
   printf("Elements of the array are:\n");
   for (i= 0; i<n; i++)
   {
     arr[i]=rand()%100;
     printf("%d ",arr[i]);
   }
   start = clock();
   for(i=0;i<=n-2;i++)
   {
      small=arr[i];
      pos=i;
      for(j=i+1;j<=n-1;j++)
      {
         if(arr[j]<small)
         {
            small=arr[j];
            pos=j;
         }
      }
      temp=arr[i];
      arr[i]=arr[pos];
      arr[pos]=temp;
   }
   end = clock();
   cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
   printf("\nSorted array is:\n");
   for(i=0;i<n;i++)
   {
      printf("%d ",arr[i]);
   }
   printf("\nExecution time for selection sort = %f seconds\n", cpu_time);
```

```
Enter the number of elements in array
500
Elements of the array are:
23 40 93 27 99 67 82 15 24 28 39 82 81 68 67 11 20 41 80 88 51 49 60 47 62 86 1 81 69 94 9 45 34 54 24 33 21 6 0 97 34 39 31 6
7 7 98 30 79 39 10 67 42 12 79 41 26 65 95 8 87 89 69 84 75 24 60 8 97 18 60 47 52 99 78 71 58 29 1 37 20 64 57 15 28 36 56 54
 54 3 14 93 44 36 77 19 12 89 79 61 7 39 8 11 90 39 82 48 68 35 38 40 51 95 55 79 83 64 86 37 67 52 30 12 88 59 83 0 0 15 14 7
 54 22 18 45 61 52 45 81 88 35 22 91 30 29 23 14 93 61 3 13 13 86 77 54 45 60 6 46 75 20 5 82 95 24 79 56 28 24 90 68 60 64 60
 42 45 83 8 39 44 12 52 9 98 29 15 95 41 22 93 69 42 99 51 89 75 82 98 3 6 88 72 18 4 84 13 49 19 21 40 15 85 92 24 35 73 40 8
3 67 14 76 36 8 27 39 98 54 21 96 10 79 84 34 50 40 18 63 89 37 36 30 52 22 74 28 9 0 20 92 67 34 21 55 43 48 46 41 55 19 37 6
5 98 73 99 0 65 69 63 6 58 52 88 62 26 63 90 35 15 63 28 34 97 49 41 40 49 87 33 4 6 22 21 56 47 72 57 12 41 72 19 51 24 7 13
2 22 56 38 37 19 18 23 16 67 64 9 68 51 42 25 9 65 46 66 12 19 75 77 60 47 96 12 24 3 77 26 78 33 16 15 52 34 39 21 53 55 30 7
4 7 24 99 16 41 45 34 6 64 9 83 77 9 31 41 85 86 18 11 64 4 80 32 8 14 23 81 20 78 63 94 37 88 93 54 81 38 40 39 55 50 74 84 1
1 5 25 96 92 95 7 8 99 39 40 60 54 15 41 26 94 5 20 31 45 13 37 78 3 78 18 10 80 92 94 91 98 19 87 42 67 46 50 18 86 43 78 92
10 72 18 4 29 90 88 74 55 77 52 10 7 22 21 87 15 15 30 65 87 69 7 54 16 9 24 54 4 55 98 15 27 68 71 56 58 59 82 13 37 86 23 44
 9 96 84 76 12 14 41 51 84 48 57 52 9 81 58 14 88 8 29 15 76
Sorted array is:
0 0 0 0 0 1 1 2 3 3 3 3 4 4 4 4 4 5 5 5 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 10 10 10 10 10 11 11
 11 11 12 12 12 12 12 12 12 13 13 13 13 13 13 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 16 16 16 16 18 18 18
 18 18 18 18 18 18 19 19 19 19 19 19 19 20 20 20 20 20 20 20 21 21 21 21 21 21 21 22 22 22 22 22 22 22 23 23 23 23 23 24 24 24 24
 24 24 24 24 24 24 25 25 25 26 26 26 26 27 27 27 28 28 28 28 28 28 29 29 29 29 29 30 30 30 30 30 30 31 31 31 32 33 33 33 34 34 34 34
 34 34 34 35 35 35 35 36 36 36 36 37 37 37 37 37 37 37 37 37 38 38 38 39 39 39 39 39 39 39 39 39 39 40 40 40 40 40 40 40 40 41 41
 41 41 41 41 41 41 41 42 42 42 42 42 42 43 43 44 44 44 44 45 45 45 45 45 45 45 45 46 46 46 46 47 47 47 47 48 48 48 49 49 49 49 50 50
 50 51 51 51 51 51 51 51 52 52 52 52 52 52 52 52 52 52 53 54 54 54 54 54 54 54 54 54 54 55 55 55 55 55 55 55 56 56 56 56 56 57 57 57
 58 58 58 58 59 59 60 60 60 60 60 60 60 60 60 61 61 61 62 62 63 63 63 63 63 64 64 64 64 64 64 65 65 65 65 65 66 67 67 67 67 67 67
 67 67 67 68 68 68 68 68 69 69 69 69 69 71 71 72 72 72 72 73 73 74 74 74 74 75 75 75 75 76 76 76 77 77 77 77 77 77 78 78 78 78
 78 78 79 79 79 79 79 79 80 80 80 81 81 81 81 81 81 81 82 82 82 82 82 82 82 83 83 83 83 83 84 84 84 84 84 84 85 85 86 86 86 86 86 86
 87 87 87 87 87 88 88 88 88 88 88 88 88 89 89 89 89 90 90 90 90 91 91 92 92 92 92 92 93 93 93 93 93 94 94 94 94 95 95 95 95 95
 96 96 96 96 97 97 97 98 98 98 98 98 98 98 99 99 99 99 99 99 99
Execution time for selection sort = 0.000419 serconds


...Program finished with exit code 0
Press ENTER to exit console.
}
```

N	TIME TAKEN
500	0.000419
1000	0.003153
1500	0.005579
2000	0.005579
2500	0.009502
3000	0.012658

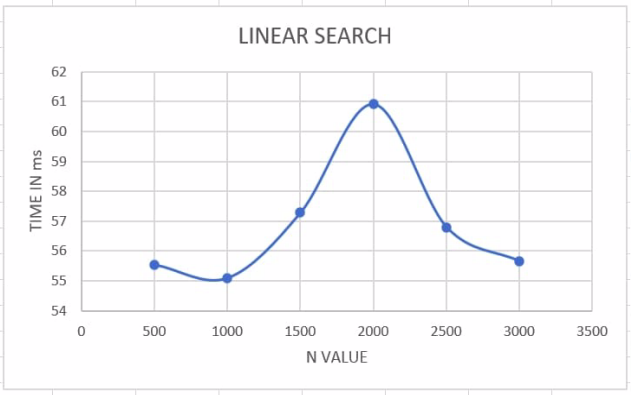



```
#include<stdio.h>

#include<time.h>




int recursiveLS( int arr[], int n, int key){

   if (arr[n]== key){

       printf("found at index %d\n\n",n);

   }

   else{

       return recursiveLS(arr,n-1,key);

   }

}

int recursiveBS( int arr[],int low,int high,int key){
```

```
    int mid = (low+high)/2;
    if(low<=high){
        if(arr[mid]==key){
        printf("Found at index %u\n\n",mid);
        }
        else if(arr[mid]<key){
            recursiveBS(arr,mid+1,high,key);
        }
        else{
           recursiveBS(arr,low,mid-1,key);
        }



    }
     else {
        return -1;
     }
}


void main(){
    int n,key,low=0;
    clock_t start,end;
    printf("Enter the no of elements\n");
    scanf("%u",&n);
    printf("Enter the element to be searched\n");
    scanf("%d",&key);
    int arr[n];
    for(int i=0;i<n;i++){
        arr[i]=i;
    }
```

```
    start = clock();
    recursiveLS(arr,n,key);
    end = clock();
    double time_taken = ((double)end-start)/CLOCKS_PER_SEC;
    printf("Time taken to search %d using linear search technique is %f
sec\n",key,time_taken);
    start = clock();
    recursiveBS(arr,low,n,key);
    end = clock();
    time_taken = ((double)end-start)/CLOCKS_PER_SEC;
    printf("Time taken to search %d using binary search technique is %f
sec\n",key,time_taken);
}
```

N value	TIME IN ms
500	55.533
1000	55.092
1500	57.284
2000	60.906
2500	56.793
3000	55.654



N value	TIME IN ms
500	53.188
1000	50.081
1500	52.353
2000	52.76
2500	56.939
3000	56.289