

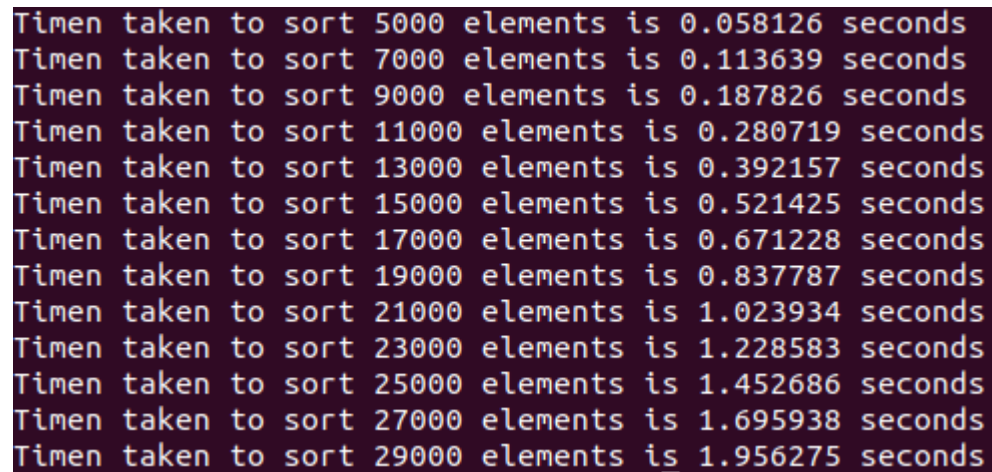
## Program\_9

Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method 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 versus n. The elements can be read from a file or can be generated using the random number generator.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
void selectionsort(int A[],int n)
{
    int i,j,min;
    for(i=0;i<n-1;i++)
    {
        min=i;
        for(j=i+1;j<n;j++)
        {
            if(A[j]<A[min])
            {
                min=j;
            }
        }
        if(min!=i)
        {
            int temp=A[j];
            A[j]=A[min];
            A[min]=temp;
        }
    }
}
int main()
{
    srand(time(NULL));
    int n=30000,i;
    int elements[n];
    for(int i=0;i<n;i++)
    {
        elements[i]=rand()%1000;
    }
    for(int size=5000;size<=n;size+=2000)
    {
        int arr[size];
        for(int i=0;i<size;i++)
        {
            arr[i]=elements[i];
        }
        clock_t start=clock();
        selectionsort(arr,size);
        clock_t stop=clock();
```

```
        double time_taken=((double)(stop-start)/CLOCKS_PER_SEC);
        printf("Timen taken to sort %d elements is %f seconds\n",size,time_taken);
    }
    return 0;
}
```

Output:



The image shows a screenshot of a terminal window with a dark background and light-colored text. It displays the output of a program that measures the time taken to sort a range of elements from 5000 to 29000 in increments of 2000. The output is a list of 15 lines, each showing the number of elements and the time taken in seconds, formatted to six decimal places. The times increase linearly as the number of elements increases.

Elements	Time (seconds)
5000	0.058126
7000	0.113639
9000	0.187826
11000	0.280719
13000	0.392157
15000	0.521425
17000	0.671228
19000	0.837787
21000	1.023934
23000	1.228583
25000	1.452686
27000	1.695938
29000	1.956275