**Slip 1**

Q.1) Write a program to sort a list of n numbers in ascending order using selection sort and determine the time required to sort the elements

// Selection sort in C

#include <stdio.h>

#include <time.h>

// function to swap the the position of two elements

void swap(int \*a, int \*b)

{

  int temp = \*a;

  \*a = \*b;

  \*b = temp;

}

void selectionSort(int array[], int size)

{

  for (int step = 0; step < size - 1; step++)

  {

    int min\_idx = step;

    for (int i = step + 1; i < size; i++)

    {

      // To sort in descending order, change > to < in this line.

      // Select the minimum element in each loop.

      if (array[i] < array[min\_idx])

        min\_idx = i;

    }

    // put min at the correct position

    swap(&array[min\_idx], &array[step]);

  }

}

// function to print an array

void printArray(int array[], int size)

{

  for (int i = 0; i < size; ++i)

  {

    printf("%d  ", array[i]);

  }

  printf("\n");

}

// driver code

int main()

{

  int data[] = {202, 124, 100, 152, 220};

  int size = sizeof(data) / sizeof(data[0]);

  clock\_t s,e,z;

  s=clock();

  selectionSort(data, size);

  printf("Sorted array in Acsending Order:\n");

  printArray(data, size);

  e=clock();

  z=e-s;

  printf("\nTime taken: %f" , (float) z/CLOCKS\_PER\_SEC);

  printf(" seconds");

}

Q.2) Write a program to sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator.

#include <iostream>

#include<time.h>

#include<stdlib.h>

#include<dos.h>

#define max 500

using namespace std;

void qsort(int[],int,int);

int partition(int [],int,int);

int main()

{

    int a[max],i,n;

    clock\_t s,e,z;

    s=clock();

    cout<<"Enter the value of n:";

    cin>>n;

    for(i=0;i<n;i++)

    a[i]=rand()%100;

    cout<<"\nThe array elements before\n";

    for(i=0;i<n;i++)

        cout<<a[i]<<"\t";

    qsort(a,0,n-1);

    cout<<"\nElements of the array after sorting are:\n";

    for(i=0;i<n;i++)

        cout<<a[i]<<"\t";

    e=clock();

    z=e-s;

    cout<<"\nTime taken:"<<z/CLOCKS\_PER\_SEC<<" seconds";

}

void qsort(int a[], int low, int high)

{

    int j;

    if(low<high)

    {

        j=partition(a,low,high);

        qsort(a, low,j-1);

        qsort(a,j+1,high);

    }

}

partition(int a[], int low, int high)

{

    int pivot,i,j,temp;

    pivot=a[low];

    i=low+1;

    j=high;

    while(1)

    {

        while(pivot>a[i] && i<=high)

            i++;

        while(pivot<a[j])

            j--;

        if(i<j)

        {

            temp=a[i];

            a[i]=a[j];

            a[j]=temp;

        }

        else

        {

            temp=a[j];

            a[j]=a[low];

            a[low]=temp;

            return j;

    }

}

}