```
#include <bits/stdc++.h>
using namespace std;

//function declarations;
void Pivot_Position(int [],int );
void Sort(int[],int,int);
void Set(int[],int [],int);
void Swap_places(int *no1,int *no2);
void Display_array(int[],int );
int choose_pivot(int,int [],int,int);
int Middle(int [],int ,int );
int partition(int [],int ,int ,int,int &,int);

int main()
{
    int n;
    cout<<"Enter the number of elements to generate in the array :\t";
    cin>>n;
    int arr[n];
    srand(time(0)); //seeding the srand method
    for(int i=0;i<n;i++)
        arr[i]=rand()%10;
    cout<<"\nThe Array generate is :   ";
    for(int i=0;i<n;i++)
        cout<<arr[i]<<" ";

    Pivot_Position(arr,n);
}
int choose_pivot(int ch,int arr[],int l,int u)
{
    int *a=&arr[l]; //lower
    int *c=&arr[u]; //upper
    int *b=&arr[(l+u)/2]; //middle
    switch(ch)
    {
        case 1:return l;
        case 2:return ((*a>=*c)?((*a<=*b)?a:((*b>=*c)?b:c)):((*c<=*b)?c:((*b>=*a)?b:a)))-arr;
        case 4:return (l+u)/2; //return middle
        default : return-1;
    }
}

int Middle(int arr[],int l,int u)
{
    int sum=0;
    for(int i=l;i<=u;i++)
        sum+=arr[i];
    return sum/(u-l+1);
}

int partition(int arr[],int l,int u,int n,int &comp,int ch)
{
    int p;
    int x;
    int j=l-1;
    if(ch!=3)
    {
      p=choose_pivot(ch,arr,l,u);
      x=arr[p];
        cout << "\n";
    }
    else
    {
        p=-1;
        x=Middle(arr,l,u);
```

```
        cout << "\n";
    }
    for(int i=l;i<=u;i++)
    {
      if(i!=p)
      {
        comp++;
       if(arr[i]<=x)
       {
          j++;
          if(j==p)p=i;
           Swap_places(&arr[i],&arr[j]);
       }
      }
    }
    if(ch!=3)Swap_places(&arr[j+1],&arr[p]);
    Display_array(arr,n);
    return(ch==3&&j==u)?j:(j+1);
}

void Sort(int arr[],int n,int ch)
{
    stack<int> Up;
    stack<int> Low;
    Up.push(n-1);
    Low.push(0);
    int comp=0;
    while(!Up.empty()&&!Low.empty())
    {
       int u=Up.top();
       int l=Low.top();
       Up.pop();
       Low.pop();
       if(u>l)
       {
         int p=partition(arr,l,u,n,comp,ch);

         Up.push(p-1);
         Low.push(l);
         Up.push(u);
         (ch!=3)?Low.push(p+1):Low.push(p);
       }
    }
    cout<<"\nTotal Comparisons :\t"<<comp;
}

void Set(int arr[],int Array[],int n)
{
  for(int i=0;i<n;i++)
        arr[i]=Array[i];
}

void Pivot_Position(int arr[],int n)
{
    int Array[n];

    for(int i=0;i<n;i++)
     Array[i]=arr[i];

    cout<<"\n\nPivot as the first Element :- \n";
    Sort(arr,n,1);

    Set(arr,Array,n);
    cout<<"\n\nPivot at Median:- \n";
    Sort(arr,n,2);

```

```
    Set(arr,Array,n);
    cout<<"\n\nPivot as mean:- \n";
    Sort(arr,n,3);

    Set(arr,Array,n);
    cout<<"\n\nPivot at the middle element :- \n";
    Sort(arr,n,4);

}

void Swap_places(int *no1,int *no2)
{
//call by reference and Swap_places the places

 int temp=*no1;
 *no1=*no2;
 *no2=temp;

}

void Display_array(int arr[],int n)
{
    for(int i=0;i<n;i++)
        cout<<arr[i]<<" ";
}
```