NAME: MOAZZAM FAROOQUI
ROLLNO: CT-24068
COURSE CODE: CT-159
ASSIGNMENT: DSA LAB#06
INSTRUCTOR: SAYYDA SAHAR FATIMA

Q1.
SOURCE CODE:

```cpp
#include<iostream>
using namespace std;

void bubblesort(int arr[],int n){
    bool swapped;
    for(int i=0;i<n-1;i++){
        swapped=false;
    for(int j=0;j<n-1-i;j++){
        if(arr[j]>arr[j+1]){
            int temp=arr[j];
            arr[j]=arr[j+1];
            arr[j+1]=temp;
            swapped=true;
        }
    }
    if(swapped==false){
        break;
    }
    }
}
```

```cpp
int maxtoys(int arr[],int n,int k){
    bubblesort(arr,n);
    int count=0;
    int sum=0;
    for(int i=0;i<n;i++){
        if(sum+arr[i]<=k){
            sum=sum+arr[i];
            count=count+1;
        }
        else{
            break;
        }
    }
        return count;
}

int main(void){
    int arr[]={1,12,5,111,200,1000,10};
    int n=7;
    int k=50;
    int result=maxtoys(arr,n,k);
    cout<<"YOU CAN BUY "<<result<<" TOYS"<<endl;
    return 0;
}
```

OUTPUT:

```
YOU CAN BUY 4 TOYS

--------------------------------
Process exited after 0.3594 seconds with return value 0
Press any key to continue . . .
```

Q2.

SOURCE CODE:

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;

int main(void){
    int n=7,k=50;
    vector<int> arr={1,12,5,111,200,1000,10};
    sort(arr.begin(),arr.end());
    int count=0,sum=0;
    for(int i=0;i<n;i++){
        if(sum+arr[i]<=k){
            sum+=arr[i];
            count++;
        }
    }
    cout<<count;
}
```

OUTPUT:

```
4
--------------------------------
Process exited after 0.03451 seconds with return value 0
Press any key to continue . . .
```

Q3.

SOURCE CODE:

```cpp
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;

class Sort{
    private:
        int swaps;
        int comparisions;
    public:
        Sort(){
            swaps=0;
            comparisions=0;
        }
        int getswaps(){
            return swaps;
        }
        int getcomparisions(){
            return comparisions;
        }
```

```cpp
        void bubblesort(int arr[],int n){
            comparisions=0;
            swaps=0;
            for(int i=0;i<n-1;i++){
                for(int j=0;j<n-1-i;j++){
                    comparisions++;
                    if(arr[j]>arr[j+1]){
                        int temp=arr[j];
                        arr[j]=arr[j+1];
                        arr[j+1]=temp;
                        swaps++;
                    }
                }
            }
        }
```

```
52          void selectionsort(int arr[],int n){
53              comparisions=0;
54              swaps=0;
55              for(int i=0;i<n-1;i++){
56                  int minindex=i;
57                  for(int j=i+1;j<n;j++){
58                      comparisions++;
59                      if(arr[j]<arr[minindex]){
60                          minindex=j;
61                      }
62                  }
63                  if(minindex!=i){
64                      int temp=arr[i];
65                      arr[i]=arr[minindex];
66                      arr[minindex]=temp;
67                      swaps++;
68                  }
69              }
70          }
71  };
```

```
73  void copyarray(int source[],int dest[],int n){
74      for(int i=0;i<n;i++){
75          dest[i]=source[i];
76      }
77  }
```

```
79  int main(void){
80      const int n=10;
81      int original[n];
82      int temp[n];
83      Sort s;
84      srand(time(0));
85      for(int i=0;i<n;i++){
86          original[i]=rand()%100;
87      }
88      cout<<"*****ORIGINAL ARRAY*****"<<endl;
89      for(int i=0;i<n;i++){
90          cout<<original[i]<<" ";
91      }
92      cout<<endl<<endl;
93
94      cout<<"Enter sorting technique(bubble / insertion / selection / all): ";
95      string choice;
96      cin>>choice;
```

```
98      int minswaps=1000;
99      int maxswaps=-1;
100     string best;
101     string worst;
102
103     if(choice=="bubble"){
104         copyarray(original,temp,n);
105         s.bubblesort(temp,n);
106         cout<<"BUBBLE SORT COMPARISIONS="<<s.getcomparisions()<<" ,SWAPS="<<s.getswaps()<<endl;
107     }
108     else if(choice=="insertion"){
109         copyarray(original,temp,n);
110         s.insertionsort(temp,n);
111         cout<<"INSERTION SORT COMPARISIONS="<<s.getcomparisions()<<" ,SWAPS="<<s.getswaps()<<endl;
112     }
113     else if(choice=="selection"){
114         copyarray(original,temp,n);
115         s.selectionsort(temp,n);
116         cout<<"SELECTION SORT COMPARISIONS="<<s.getcomparisions()<<" ,SWAPS="<<s.getswaps()<<endl;
117     }
```

```
    else if(choice=="all"){
        copyarray(original,temp,n);
        s.bubblesort(temp,n);
        int bubblec=s.getcomparisions();
        int bubbles=s.getswaps();
        cout<<"BUBBLE SORT COMPARISIONS="<<bubblec<<" ,SWAPS="<<bubbles<<endl;
        if(bubbles<minswaps){minswaps=bubbles;best="BUBBLE SORT";}
        if(bubbles>maxswaps){maxswaps=bubbles;worst="BUBBLE SORT";}

        copyarray(original,temp,n);
        s.insertionsort(temp,n);
        int insertc=s.getcomparisions();
        int inserts=s.getswaps();
        cout<<"INSERTION SORT COMPARISIONS="<<insertc<<" ,SWAPS="<<inserts<<endl;
        if(inserts<minswaps){minswaps=inserts;best="INSERTION SORT";}
        if(inserts>maxswaps){maxswaps=inserts;worst="INSERTION SORT";}

        copyarray(original,temp,n);
        s.selectionsort(temp,n);
        int selectionc=s.getcomparisions();
        int selections=s.getswaps();
        cout<<"SELECTION SORT COMPARISIONS="<<selectionc<<" ,SWAPS="<<selections<<endl;
        if(selections<minswaps){minswaps=selections;best="SELECTION SORT";}
        if(selections>maxswaps){maxswaps=selections;worst="SELECTION SORT";}
```

```
            cout<<"BEST TECHNIQUE: "<<best<<endl;
            cout<<"WORST TECHNIQUE: "<<worst<<endl;
        }
        else{
            cout<<"INVALID CHOICE"<<endl;
        }
        return 0;
}
```

OUTPUT:

```
*****ORIGINAL ARRAY*****
93 8 53 74 68 97 48 32 19 33

ENTER SORTING TECHNIQUE(BUBBLE/INSERTION/SELECTION/ALL):all
BUBBLE SORT COMPARISIONS=45 ,SWAPS=29
INSERTION SORT COMPARISIONS=38 ,SWAPS=29
SELECTION SORT COMPARISIONS=45 ,SWAPS=7
BEST TECHNIQUE: SELECTION SORT
WORST TECHNIQUE: BUBBLE SORT

--------------------------------
Process exited after 4.934 seconds with return value 0
Press any key to continue . . .
```

Q4.

SOURCE CODE:

```cpp
#include<iostream>
using namespace std;

void flip(int arr[],int k){
    int start=0;
    int end=k-1;
    while(start<end){
        int temp=arr[start];
        arr[start]=arr[end];
        arr[end]=temp;
        start++;
        end--;
    }
}

int findMaxIndex(int arr[],int n){
    int maxIndex=0;
    for(int i=1;i<n;i++){
        if(arr[i]>arr[maxIndex]){
            maxIndex=i;
        }
    }
    return maxIndex;
}
```

```cpp
void pancakeSort(int arr[],int n){
    for(int size=n;size>1;size--){
        int maxIndex=findMaxIndex(arr,size);
        if(maxIndex!=size-1){
            flip(arr,maxIndex+1);
            flip(arr,size);
        }
    }
}

int main(void){
    int arr[]={3,2,4,1};
    int n=4;
    pancakeSort(arr,n);
    cout<<"FINAL OUTPUT:";
    for(int i=0;i<n;i++){
        cout<<arr[i]<<" ";
    }
    return 0;
}
```

OUTPUT:

```
FINAL OUTPUT:1 2 3 4
--------------------------------
Process exited after 0.3064 seconds with return value 0
Press any key to continue . . .
```

Q5.

SOURCE CODE:

```cpp
#include<iostream>
using namespace std;

void sortcolors(int nums[], int n){
    int low=0;
    int mid=0;
    int high=n-1;

    while(mid<=high){
        if(nums[mid]==0){
            int temp=nums[low];
            nums[low]=nums[mid];
            nums[mid]=temp;
            low++;
            mid++;
        }
        else if(nums[mid]==1){
            mid++;
        }
        else{
            int temp=nums[mid];
            nums[mid]=nums[high];
            nums[high]=temp;
            high--;
        }
    }
}
```

```cpp
int main(void){
    int nums1[]={2,0,2,1,1,0};
    int nums2[]={2,0,1};
    int n1=6;
    int n2=3;

    sortcolors(nums1,n1);
    sortcolors(nums2,n2);


    cout<<"FIRST ARRAY:"<<endl;
    for(int i=0;i<n1;i++){
        cout<<nums1[i]<<" ";;
    }
    cout<<endl;
    cout<<"SECOND ARRAY:"<<endl;
    for(int j=0;j<n2;j++){
        cout<<nums2[j]<<" ";
    }
    return 0;
}
```

OUTPUT:

```
FIRST ARRAY:
0 0 1 1 2 2
SECOND ARRAY:
0 1 2
--------------------------------
Process exited after 0.5436 seconds with return value 0
Press any key to continue . . .
```