LAB 6 SORTING

1. WAP to implement a function Rdm(n) which returns an array of random numbers{between 0 to 99}, where n is the size of array. (Hint: use dynamic memory allocation concept)

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int* Rdm(int n) {
        int* randomArray = new int[n];
        srand(time(0));

        for (int i = 0; i < n; i++) {
        randomArray[i] = std::rand() % 100;
        }
        return randomArray;
}

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

2. WAP to implement the bubble sort and show the output of each pass

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include <iostream>
#include "rdm.h"
using namespace std;

int main(){
        int n = 10;
        int *arr = Rdm(n);
```

```
        cout << "Before Swap!!" << endl;
        display(arr, n);
        cout << "Sorting: "<< endl;
        for(int i = 0; i<n-1; i++){
        for(int j = 0; j < n-i-1; j++){
        if(arr[j] > arr[j+1]){
                int tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
        }
        }
        cout << "After pass " << i << ": ";
        display(arr, n);
        }
        delete[] arr;
}
```

3. WAP to implement the selection sort and show the output of each pass.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include <iostream>
#include "rdm.h"
using namespace std;

int main(){
        int n = 10;
        int *arr = Rdm(n);
        cout << "Befor sorting!!: ";
        display(arr, n);
        for(int i = 0; i<n-1; i++){
        int min = i;
        for(int j= i+1; j<n; j++){
        min = j;
        }
        int tmp = arr[i];
        arr[i] = arr[min];
        arr[min] = tmp;
        cout << "After swap " << i << " : ";
        display(arr, n);
```

```
        }
        delete[] arr;
}
```

4. WAP to implement the insertion sort and show the output of each pass.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
/*************************************************************/
#include <iostream>
#include "rdm.h"
using namespace std;


int main(){
        int n = 6;
        int *arr = Rdm(n);
        cout << "Before sort!!"<< endl;
        display(arr, n);
        for(int i = 1; i<n; i++){
        int key = arr[i];
        int j = i -1;

        while (j>=0 && arr[j] > key){
        arr[j+1] = arr[j];
        j--;
        }
        arr[j+1] = key;
        cout << "After swap " << i -1<< " : ";
        display(arr, n);
        }
        delete[] arr;
}
```

5. WAP to implement the quick sort and show the output of each pass.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
/*************************************************************/
#include <iostream>
#include "rdm.h"
using namespace std;
```

```
static int a = 0;
int Partition(int arr[], int left, int right) {
        int pivot = arr[right];
        int i = left - 1;

        for (int j = left; j < right; j++) {
        if (arr[j] <= pivot) {
        i++;
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[right];
        arr[right] = temp;
        return i + 1;
}

void QuickSort(int arr[], int n, int left, int right) {
        if (left < right) {
        int pivotIndex = Partition(arr, left, right);
        cout << "After pass ";
        QuickSort(arr,n, left, pivotIndex - 1);
        QuickSort(arr, n,pivotIndex + 1, right);
        cout << "After pass " << a++ << " : ";
        display(arr,n);
        }
}

int main() {
        int n = 11;
        int* array = Rdm(n);

        cout << "Unsorted Array: ";
        display(array, n);
        cout << endl;
        QuickSort(array,n , 0, n - 1);
        cout << endl;
```

```
	delete[] array;
	return 0;
}
```

6. WAP to implement the merge sort and show the output of each pass.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include<iostream>
#include<stdlib.h>
#include<cmath>
#include "rdm.h"
using namespace std;
void merge(int a[],int low,int mid1,int mid2,int high){
	int c=0;
	int left=low;
	int right=mid2;
	int temp[high-low+1];
	while((left<=mid1) && (right<=high)){
		if(a[left]<=a[right]){
			temp[c]=a[left];
			left=left+1;
		}
		else{
			temp[c]=a[right];
			right=right+1;
		}
		c=c+1;
	}
	while(left<=mid1){
		temp[c]=a[left];
		left=left+1;
		c=c+1;
	}
	while(right<=high){
		temp[c]=a[right];
		right=right+1;
		c=c+1;
	}
```

```
        for(int i=low,j=0;i<=high;j++,i++){
                a[i]=temp[j];
        }
}
void merge_sort(int a[],int low,int high){
        int mid;
        if(low<high){
                cout<<"low ="<<low<<" High="<<high<<endl;
                mid=ceil((low+high)/2.0);
                merge_sort(a,low,mid-1);
                merge_sort(a,mid,high);
                merge(a,low,mid-1,mid,high);
        }
}
int main(){
        cout<<"enter size ";
        int n;
        cin>>n;
        int *arr;
        arr=random(n);
        for(int i=0;i<n;i++){
                cout<<arr[i]<<" ";
        }
        cout<<"\nn is ="<<n;
        merge_sort(arr,0,n-1);
        cout<<endl;
        for(int i=0;i<n;i++){
                cout<<arr[i]<<" ";
        }
                return 0;
}
```

7. WAP to sort a character array using insertion sort in alphabetic order and print number of shifts.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include<iostream>
using namespace std;
int main(){
```

```
        char arr[10];
        for(int i=0;i<10;i++){
                cin>>arr[i];
        }
        int count=0;
        for(int i=1;i<10;i++){
                int key=arr[i];
                int j=i-1;
                while(j>=0 && arr[j]>key){
                        arr[j+1]=arr[j];
                        j=j-1;
                }
                arr[j+1]=key;
                count++;
        }
        for(int i=0;i<10;i++){
                cout<<arr[i];
        }
        cout<<endl<<"total shift = "<<count<<endl;
        return 0;
}
```

398. WAP to insert an element in sorted array and after insertion order should not change.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
/*************************************************************/
#include<iostream>
using namespace std;
int main(){
        cout<<"enter initial size of an array : "<<endl;
        int n;
        cin>>n;
        int arr[100]={0};
        cout<<"enter array"<<endl;
        for(int i=0;i<n;i++){
                cin>>arr[i];
        }
        char p = 'y';
        while(p!='n'){
```

```
            cout<<endl<<"enter element you want to enter ";
            int ny;
            cin>>ny;
            arr[n]=ny;
            n=n+1;
            for(int i=1;i<n;i++){
                  int key=arr[i];
                  int j=i-1;
                  while(j>=0 && arr[j]>key){
                        arr[j+1]=arr[j];
                        j=j-1;
                  }
                  arr[j+1]=key;
            }
            cout<<endl<<"do you want to enter a element ";
            cin>>p;
      }
      for(int i=0;i<n;i++){
            cout<<arr[i]<<" ";
      }
      return 0;
}
```

9. WAP to implement stable selection sort.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
/*************************************************************/
#include <iostream>
using namespace std;

void stableSelectionSort(int arr[], int n) {
      for (int i = 0; i < n - 1; ++i) {
            int minIndex = i;
            for (int j = i + 1; j < n; ++j) {
                  if (arr[j] < arr[minIndex]) {
                        minIndex = j;
                  }
            }
            int minValue = arr[minIndex];
            for (int k = minIndex; k > i; --k) {
```

```
                arr[k] = arr[k - 1];
            }
            arr[i] = minValue;
        }
}

int main() {
        int arr[] = {64, 25, 12, 22, 11};
        int n = sizeof(arr) / sizeof(arr[0]);
        cout << "Original array: ";
        for (int i = 0; i < n; ++i) {
            cout << arr[i] << " ";
        }
        cout << endl;
        stableSelectionSort(arr, n);
        cout << "Sorted array: ";
        for (int i = 0; i < n; ++i) {
            cout << arr[i] << " ";
        }
        cout << endl;
        return 0;
}
```

10. WAP to implement online insertion sort such that it can sort the numbers entered during the execution of the program.

```
/*************************************************************
//This program is developed by Aman Singh Rawat(221B056)
*************************************************************/
#include<iostream>
using namespace std;

int main(){
        int n=0;
        int arr[100]={0};
        cout<<"-100 for exit from online sorting system "<<endl;
        int ny;
        cin>>ny;
        while(ny!= -1){
            arr[n]=ny;
            n=n+1;
```

```
            for(int i=1;i<n;i++){
                int key=arr[i];
                int j=i-1;
                while(j>=0 && arr[j]>key){
                    arr[j+1]=arr[j];
                    j=j-1;
                }
                arr[j+1]=key;
            }
            for(int i=0;i<n;i++){
                cout<<arr[i]<<" ";
            }
            cout<<endl;
            cin>>ny;
        }
        return 0;
}
```