**DSA LAB ASSIGNMENT SUBMISSION**

Name : Vanshikaa

Roll No. : 102103580

Group : 2CO21

**LAB 1 :**

Q1 a

#include <iostream>

using namespace std;

void swap(int \*xp, int \*yp)

{

    int temp = \*xp;

    \*xp = \*yp;

    \*yp = temp;

}

void selectionSort(int arr[], int n)

{

    int i, j, min\_idx;

    for (i = 0; i < n-1; i++)

    {

        min\_idx = i;

        for (j = i+1; j < n; j++)

        if (arr[j] < arr[min\_idx])

            min\_idx = j;

        if(min\_idx!=i)

            swap(&arr[min\_idx], &arr[i]);

    }

}

void printArray(int arr[], int size)

{

    int i;

    for (i=0; i < size; i++)

        cout << arr[i] << " ";

    cout << endl;

}

int main()

{

    int arr[] = {64, 25, 12, 22, 11};

    int n = sizeof(arr)/sizeof(arr[0]);

    selectionSort(arr, n);

    cout << "Sorted array: \n";

    printArray(arr, n);

    return 0;

}

Q1 b.

#include<iostream>

using namespace std;

int main(){

    int \*arr;

    int n,temp;

    cout<<"enter the size : "<<endl;

    cin>>n;

    arr = new int[n];

    cout<<"enter the elements"<<endl;

    for(int i = 0;i<n;i++){

        cin>>arr[i];

    }

    for(int i = 1;i<n;i++){

        temp = arr[i];

        int j = i;

        while(j>0 && arr[j-1]>temp){

            arr[j] = arr[j-1];

            j = j-1;

        }

        arr[j] = temp;

    }

    cout<<"sorted array"<<endl;

    for(int i = 0;i<n;i++){

        cout<<arr[i]<<" ";

    }

}

Q1 c.

#include<iostream>

using namespace std;

int main(){

    int arr[5];

    cout<<"enter elements : "<<endl;;

    for (int i = 0; i < 5; i++)

    {

       cin>>arr[i];

    }

    for(int i = 0;i<4;i++){

        int flag = 0;

        for (int j = 0; j < 4-i; j++)

        {

            if(arr[j]>arr[j+1]){

                flag = 1;

                swap(arr[j],arr[j+1]);

            }

            if (flag = 0){

                break;

            }

        }

    }

    cout<<"after bubble sort : "<<endl;

    for (int i = 0; i <5; i++)

    {

    cout<<arr[i]<<" ";

    }

}

Q1 d.

#include <iostream>

using namespace std;

void Merge(int \*arr, int low, int mid, int high)

{

    int \*T;

    int ac = low;

    int bc = mid;

    int tc = 0;

    int l = mid - 1;

    while (ac <= l && bc <= high)

    {

        if (arr[ac] <= arr[bc])

        {

            T[tc] = arr[ac];

            tc++;

            ac++;

        }

        else

        {

            T[tc] = arr[bc];

            tc++;

            bc++;

        }

    }

    while (ac <= l)

    {

        T[tc] = arr[ac];

        tc++;

        ac++;

    }

    while (bc <= high)

    {

        T[tc] = arr[bc];

        tc++;

        bc++;

    }

    for (int i = low; i < high; i++)

    {

        arr[i] = T[i - low];

    }

}

void MergeSort(int \*arr, int low, int high)

{

    int mid;

    if (low < high)

    {

        mid = (low + high) / 2;

        MergeSort(arr, low, mid);

        MergeSort(arr, mid + 1, high);

        Merge(arr, low, mid + 1, high);

    }

}

int main()

{

    int n;

    int arr[100];

    cout << "enter size of array" << endl;

    cin >> n;

    cout << "enter the elements" << endl;

    for (int i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

    MergeSort(arr, 0, n - 1);

    for (int i = 0; i < n; i++)

    {

        cout << arr[i] << " ";

    }

}

Q1 e.

#include <iostream>

using namespace std;

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

{

    int t = \*a;

    \*a = \*b;

    \*b = t;

}

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

{

    int pivot = arr[high];

    int i

        = (low - 1);

    for (int j = low; j <= high - 1; j++) {

        if (arr[j] < pivot) {

            i++;

            swap(&arr[i], &arr[j]);

        }

    }

    swap(&arr[i + 1], &arr[high]);

    return (i + 1);

}

void quickSort(int arr[], int low, int high)

{

    if (low < high) {

        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);

        quickSort(arr, pi + 1, high);

    }

}

void printArray(int arr[], int size)

{

    int i;

    for (i = 0; i < size; i++)

        cout << arr[i] << " ";

    cout << endl;

}

int main()

{

    int arr[] = { 10, 7, 8, 9, 1, 5 };

    int n = sizeof(arr) / sizeof(arr[0]);

    quickSort(arr, 0, n - 1);

    cout << "Sorted array: \n";

    printArray(arr, n);

    return 0;

}

Q1 f.

#include<iostream>

using namespace std;

int getMax(int a[], int n) {

   int max = a[0];

   for(int i = 1; i<n; i++) {

      if(a[i] > max)

         max = a[i];

   }

   return max;

}

void countSort(int a[], int n)

{

   int output[n+1];

   int max = getMax(a, n);

   int count[max+1];

  for (int i = 0; i <= max; ++i)

  {

    count[i] = 0;

  }

  for (int i = 0; i < n; i++)

  {

    count[a[i]]++;

  }

   for(int i = 1; i<=max; i++)

      count[i] += count[i-1];

  for (int i = n - 1; i >= 0; i--) {

    output[count[a[i]] - 1] = a[i];

    count[a[i]]--;

}

   for(int i = 0; i<n; i++) {

      a[i] = output[i];

   }

}

void printArr(int a[], int n)

{

    int i;

    for (i = 0; i < n; i++)

        printf("%d ", a[i]);

}

int main() {

    int a[] = { 11, 30, 24, 7, 31, 16 };

    int n = sizeof(a)/sizeof(a[0]);

    printf("Before sorting array elements are - \n");

    printArr(a, n);

    countSort(a, n);

    printf("\nAfter sorting array elements are - \n");

    printArr(a, n);

    return 0;

}