**Module 7**

#include <iostream> **Topic 1**

using namespace std;

void insertion(int a[], int n)

{

    int i = 1, j, k;

    while (i < n)

    {

        k = a[i];

        j = i - 1;

        cout << "\nkey = " << k << endl;

        // cout << k << " ";

        while (j >= 0 && a[j] > k)

        {

            a[j + 1] = a[j];

            // cout << a[j] << " ";

            j--;

        }

        a[j + 1] = k;

        i++;

    }

}

void print(int a[], int n)

{

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    cout << endl;

}

int main()

{

    int n;

    cout << "Enter your array size : ";

    cin >> n;

    int a[n];

    cout << "Enter your array elements = ";

    for (int i = 0; i < n; i++)

        cin >> a[i];

    cout << "Before Sorting " << endl;

    print(a, n);

    insertion(a, n);

    cout << "After Sorting" << endl;

    print(a, n);}

#include <iostream>

using namespace std; **Topic 2**

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

{

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void display(int array[], int size)

{

    for (int i = 0; i < size; i++)

    {

        cout << array[i] << " ";

    }

    cout << endl;

}

void selection(int a[], int n)

{

    for (int i = 0; i < n - 1; i++)

    {

        int j, min = i;

        for (j = i + 1; j < n; j++)

        {

            if (a[j] < a[min])

                min = j;

        }

        cout << "min = " << a[min] << " "

             << "swap position = " << i << endl;

        if (min != i)

            swap(&a[min], &a[i]);

        cout << "min = " << a[min] << " "

             << "swap position = " << i << endl;

        cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    }

}

int main()

{

    int i = 0, n = 5;

    int a[n];

    cout << "Enter your values : ";

    while (i < n)

    {

        cin >> a[i];

        i++;

    }

    cout << "Before sorting" << endl;

    display(a, n);

    selection(a, n);

    cout << "After Sorting" << endl;

    display(a, n);

}

#include <iostream> **Topic 3**

using namespace std;

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

{

    int t = \*a;

    \*a = \*b;

    \*b = t;

}

int partioner(int a[], int x, int y)

{

    int p = a[y];

    cout << "Pivot : " << p << endl;

    int j = x, i = x - 1;

    while (j <= y - 1)

    {

        if (a[j] < p)

        {

            i++;

            swap(&a[i], &a[j]);

        }

        j++;

    }

    swap(&a[i + 1], &a[y]);

    return i + 1;

}

void qs(int m[], int size, int x, int y)

{

    if (x < y)

    {

        int h = partioner(m, x, y);

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

        {

            if (i == h)

                cout << "(" << m[i] << ")";

            else

            {

                cout << m[i] << " ";

            }

        }

        cout<< endl;

        qs(m, size, x, h - 1);

        qs(m, size, h + 1, y);

    }

}

int main()

{

    int a[] = {2, 5, 1, 7, 8, 3, 9, 4};

    int s = sizeof(a) / sizeof(a[0]);

    // cout << s << endl;

    cout << "Before Swap" << endl;

    for (int i : a)

        cout << i << " ";

    cout << endl;

    qs(a, s, 0, s - 1);

    cout << "After Swap" << endl;

    for (int i : a)

        cout << i << " ";

    cout << endl;

}

#include <iostream> **Topic 4**

using namespace std;

void merger(int a[], int x, int m, int y)

{

    int i = 0, j = 0;

    auto size1 = m - x + 1, size2 = y - m;

    int \*b = new int[size1], \*c = new int[size2];

    cout << "Low = " << x << " "

         << "Mid= " << m << " End = " << y << " Size1 = " << size1 << " Size2 = " << size2 << endl;

    while (i < size1)

    {

        b[i] = a[x + i];

        cout << b[i] << " ";

        i++;

    }

    while (j < size2)

    {

        c[j] = a[m + 1 + j];

        cout << c[j] << " ";

        j++;

    }

    cout << endl;

    auto ii = 0, jj = 0, kk = x;

    while (ii < size1 && jj < size2)

    {

        if (b[ii] <= c[jj])

        {

            a[kk] = b[ii];

            cout << b[ii] << " ";

            ii++;

        }

        else

        {

            a[kk] = c[jj];

            cout << c[jj] << " ";

            jj++;

        }

        kk++;

    }

    while (ii < size1)

    {

        a[kk] = b[ii];

        cout << b[ii] << " ";

        ii++;

        kk++;

    }

    while (jj < size2)

    {

        a[kk] = c[jj];

        cout << c[jj] << " ";

        jj++;

        kk++;

    }

    cout << endl;

    delete[] b;

    delete[] c;

}

void mergeSort(int a[], int x, int y)

{

    auto m = x + (y - x) / 2;

    if (x < y)

    {

        mergeSort(a, x, m);

        mergeSort(a, m + 1, y);

        merger(a, x, m, y);

    }

}

void print(int a[], int x, int y)

{

    for (auto i = x; i < y; i++)

        cout << a[i] << " ";

    cout << endl;

}

int main()

{

    int a[] = {10, 78, 32, 90, 20, 19, 25, 25};

    auto s = sizeof(a) / sizeof(a[0]);

    cout << "Before Sorting" << endl;

    print(a, 0, s);

    mergeSort(a, 0, s - 1);

    cout << "After Sorting" << endl;

    print(a, 0, s);

}