Name: Shah Siddh Tejaskuamr

Reg.No: 20BCE1937

Merge And Quick Sort Program

---------------------------------------------------------------------------------------------------------------------------------

#include <iostream>

using *namespace* std;

*void* printArray(*int* \**A*, *int* *n*)

{

    for (*int* i = 0; i < *n*; i++)

    {

        printf("%d ", *A*[i]);

    }

    printf("\n");

}

*void* Swap(*int* *x*, *int* *y*)

{

*int* temp;

    temp = *x*;

*x* = *y*;

*y* = temp;

}

*int* partition\_front(*int* *A*[], *int* *low*, *int* *high*)

{

*int* pivot = *A*[*low*];

*int* i = *low* + 1;

*int* j = *high*;

*int* temp;

    do

    {

        while (*A*[i] <= pivot)

        {

            i++;

        }

        while (*A*[j] > pivot)

        {

            j--;

        }

        if (i < j)

        {

            Swap(*A*[i], *A*[j]);

        }

    } while (i < j);

    Swap(*A*[*low*], *A*[j]);

    return j;

}

*int* partition\_mid(*int* *A*[], *int* *low*, *int* *high*)

{

*int* pivot = (*high* + *low*) / 2;

*int* i = *low*, j = *high*;

    while (i < j)

    {

        while (*A*[i] < pivot)

        {

            i++;

        }

        while (*A*[j] > pivot)

        {

            j--;

        }

        Swap(*A*[i], *A*[j]);

    }

    Swap(*A*[j], pivot);

    return j;

}

*int* partition\_back(*int* *A*[], *int* *low*, *int* *high*)

{

*int* pivot = *A*[*low*];

*int* i = *high* - 1;

*int* j = *low*;

*int* temp;

    do

    {

        while (*A*[i] > pivot)

        {

            i++;

        }

        while (*A*[j] < pivot)

        {

            j--;

        }

        Swap(*A*[i], *A*[j]);

    } while (i < j);

    // Swap A[low] and A[j]

    Swap(*A*[*high* - 1], *A*[j]);

    return j;

}

*void* quickSort(*int* *A*[], *int* *low*, *int* *high*)

{

*int* partitionIndex; // Index of pivot after partition

*char* s;

    cout << "Pivot: a.front b.middle c.back ";

    cin >> s;

    if (s == 'a')

    {

        if (*low* < *high*)

        {

            partitionIndex = partition\_front(*A*, *low*, *high*);

            quickSort(*A*, *low*, partitionIndex - 1);  // sort left subarray

            quickSort(*A*, partitionIndex + 1, *high*); // sort right subarray

        }

    }

    else if (s == 'c')

    {

        if (*low* < *high*)

        {

            partitionIndex = partition\_back(*A*, *low*, *high*);

            quickSort(*A*, *low*, partitionIndex - 1);  // sort left subarray

            quickSort(*A*, partitionIndex + 1, *high*); // sort right subarray

        }

    }

    else if (s == 'b')

    {

        if (*low* < *high*)

        {

            partitionIndex = partition\_mid(*A*, *low*, *high*);

            quickSort(*A*, *low*, partitionIndex - 1);  // sort left subarray

            quickSort(*A*, partitionIndex + 1, *high*); // sort right subarray

        }

    }

}

*void* merge(*int* *arr*[], *int* *low*, *int* *mid*, *int* *high*)

{

*int* i = *low*, j = *mid* + 1, k = *low*;

*int* temp[*high*];

    while (i<=*mid* && j<=*high*)

    {

        if (*arr*[i] < *arr*[j])

        {

            temp[k] = *arr*[i];

            i++;

        }

        else

        {

            temp[k] = *arr*[j];

            j++;

        }

        k++;

    }

    if (i>*mid* && j<=*high*)

    {

        while (j<=*high*)

        {

            temp[k] = *arr*[j];

            k++; j++;

        }

    }

    else if (j>*high* && i<=*mid*)

    {

        while (i<=*mid*)

        {

            temp[k] = *arr*[i];

            k++; i++;

        }

    }

    for (*int* x=*low* ; x<*high*; x++)

    {

*arr*[x] = temp [x];

    }

}

*void* mergeSort(*int* *arr*[], *int* *low*, *int* *high*)

{

*int* mid;

    cout << "Merge Sort called " << endl;

    if (*low* < *high*)

    {

        mid = (*low* + *high*) / 2;

        mergeSort(*arr*, *low*, mid);

        mergeSort(*arr*, mid + 1, *high*);

        merge(*arr*, *low*, mid, *high*);

    }

}

*int* main()

{

*int* n;

    cout << "Enter size: ";

    cin >> n;

*int* arr[n];

    cout << "Enter the ele: ";

    for (*int* i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

*int* choice;

    cout << "SELECT: 1.Quick\_Sort 2.Merge\_Sort: ";

    cin >> choice;

    if (choice == 1)

    {

        quickSort(arr, 0, n - 1);

        printArray(arr, n);

    }

    else if (choice == 2)

    {

        mergeSort(arr, 0, n - 1);

        printArray(arr, n);

    }

    return 0;

}