#include <stdio.h>

#include <stdlib.h>

// SORTING ALGORITHMS

int BubbleSort()

{

int arr[]={5,67,82,3,91,22,18};

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

BubbleSort(arr, n);

int i, j;

for (i = 0; i < n - 1; i++)

{

for (j = 0; j < n - i - 1; j++)

{

if (arr[j] > arr[j + 1])

{

int\* xp;

int\* yp;

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

}

}

int size;

for (i = 0; i < size; i++)

{

printf("Sorted array: \n");

printf("%d \n", arr[i]);

}

return 0;

}

int insertionSort()

{

int arr[] = { 12, 11, 13, 5, 6 };

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

int i, key, j;

for (i = 1; i < n; i++) {

key = arr[i];

j = i - 1;

/\* Move elements of arr[0..i-1], that are

greater than key, to one position ahead

of their current position \*/

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

//Print

for (i = 0; i < n; i++)

{

printf("%d \n", arr[i]);

}

return 0;

}

int mergesort()

{

int arr[] = { 5, 11, 22, 16, 9, 7 };

int arr\_size = sizeof(arr) / sizeof(arr[0]);

printf("Given array is \n");

int i, size;

int A[10];

for (i = 0; i < size; i++)

{

printf("%d \n", A[i]);

}

int left;

int middle;

int right;

int j, k;

int n1 = middle - left + 1;

int n2 = right - middle;

int L[n1], R[n2];

for (i = 0; i < n1; i++)

{

L[i] = arr[left + i];

}

for (j = 0; j < n2; j++)

{

R[j] = arr[middle + 1 + j];

}

i = 0;

j = 0;

k = left;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

//SORTING PORTION

if (left < right) {

int middle = left + (right - left) / 2;

// Sort first and second halves

mergesort(arr, left, middle);

mergesort(arr, middle + 1, right);

mergesort(arr, left, middle, right);

}

for (i = 0; i < size; i++)

printf("%d ", A[i]);

printf("\n");

return 0;

}

void selectionsort()

{

int \*a;

int \*b;

int temp = \*a;

\*a = \*b;

\*b = temp;

int array[10];

int data[] = {20, 12, 10, 15, 2};

int array\_size = array\_sizeof(data) / array\_sizeof(data[0]);

for (int move = 0; move < array\_size - 1; move++) {

int min\_index = move;

for (int i = move + 1; i < array\_size; i++) {

if (array[i] < array[min\_index])

min\_index = i;

}

swap(&array[min\_index], &array[move]);

}

for (int i = 0; i < array\_size; ++i) {

printf("Sorted array in ascending order:");

printf("%d ", array[i]);

}

printf("\n");

}

int quicksort(){

int num[25];

int first;

int last;

int i=0;

int j, turn, temp;

if(first<last){

turn=first;

i=first;

j=last;

while(i<j){

while(num[i]<=num[turn]&&i<last)

i++;

while(num[j]>num[turn])

j--;

if(i<j){

temp=num[i];

num[i]=num[j];

num[j]=temp;

}

}

temp=num[turn];

num[turn]=num[j];

num[j]=temp;

quicksort(num,first,j-1);

quicksort(num,j+1,last);

}

int count;

printf("Display:");

printf("Number of elements: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

for(i=0;i<count;i++)

{

scanf("%d",&num[i]);

}

quicksort(num,0,count-1);

printf("Sorted elements are: ");

for(i=0;i<count;i++)

{

printf(" %d",num[i]);

}

return 0;

}

int heapsort()

{

int arr[100],n,temp,last,i;

clrscr();

printf("How many Numbers you want to enter in your array: \n");

scanf("%d",&n);

printf("Enter Elements in array:\n");

for(i=1;i<=n;i++)

scanf("%d",&arr[i]);

arr[0]=n;

for(i=n/2;i>=1;i--)

{

int j,temp,k=1;

n=arr[0];

while(2\*i<=n && k==1)

{

j=2\*i;

if(j+1<=n && arr[j+1] > arr[j])

j=j+1;

if( arr[j] < arr[i])

k=0;

else

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

i=j;

}

}

}

while(arr[0] > 1)

{

last=arr[0];

temp=arr[1];

arr[1]=arr[last];

arr[last]=temp;

arr[0]--;

int j,temp,n,k=1;

n=arr[0];

while(2\*i<=n && k==1)

{

j=2\*i;

if(j+1<=n && arr[j+1] > arr[j])

j=j+1;

if( arr[j] < arr[i])

k=0;

else

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

i=j;

}

}

}

printf("Array After Heap Sort\n");

for(i=1;i<=n;i++)

printf("%d ",arr[i]);

getchar();

}

int main()

{

int noofnodes;

struct node \*Firstnode;

Firstnode = NULL;

printf("Input the num of elements in the linked list: ");

scanf("%d", &noofnodes);

BubbleSort();

insertionSort();

mergesort();

selectionsort();

quicksort();

heapsort();

return 0;

}