#include <stdio.h>

#include <string.h>

#include <time.h>

#include <stdlib.h>

//insertion Sort Algorithm

void insertionSort(int arr[], int n)

{

int i, key, j;

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

key = arr[i];

j = i - 1;

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

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

j = j - 1;

}

arr[j + 1] = key;

}

}

// Char array insertion sort

void insertionSortChar(char arr[], int n)

{

int i, j;

char key;

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

key = arr[i];

j = i - 1;

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

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

j = j - 1;

}

arr[j + 1] = key;

}

}

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

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void swapChar(char \*xp, char \*yp)

{

char temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

//Selection Algorithm

void selectionSort(int arr[], int n)

{

int i, j, min;

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

{

min = i;

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

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

min = j;

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

}

}

void selectionSortChar(char arr[], int n)

{

int i, j;

char min;

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

{

min = i;

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

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

min = j;

swapChar(&arr[min], &arr[i]);

}

}

// Bubble Sort Algorithm

void bubbleSort(int arr[], int 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])

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

}

void bubbleSortChar(char arr[], int 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])

swapChar(&arr[j], &arr[j+1]);

}

// Merge sort algorithm

void merge(int arr[], int l, int m, int r)

{

int i, j, k, n1, n2;

n1 = m - l + 1;

n2 = r - m;

int L[n1], R[n2];

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

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

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

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

i = 0;

j = 0;

k = l;

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++;

}

}

void mergeSort(int arr[], int l, int r)

{

if (l < r) {

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

void mergeChar(char arr[], int l, int m, int r)

{

int i, j, k, n1, n2;

n1 = m - l + 1;

n2 = r - m;

int L[n1], R[n2];

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

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

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

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

i = 0;

j = 0;

k = l;

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++;

}

}

void mergeSortChar(char arr[], int l, int r)

{

if (l < r) {

int m = l + (r - l) / 2;

mergeSortChar(arr, l, m);

mergeSortChar(arr, m + 1, r);

mergeChar(arr, l, m, r);

}

}

void printArray(int arr[], int n)

{

int i;

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

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

printf("\n");

}

int main() {

//Selection Sort

clock\_t t;

//My school number 2015556451

int no[] = { 2, 0, 1, 5, 5, 5, 6, 4, 5, 1 };

//My name

char name[] = "enver atmaca";

int n = sizeof(no) / sizeof(no[0]);

int charSize = strlen(name);

t = clock();

insertionSort(no, n);

t = clock() - t;

printArray(no, n);

printf("insertion sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

//start = end;

insertionSortChar(name, charSize);

//end = clock();

t = clock() - t;

printf("%s\n", name);

//printf("insertion char sort time: %f\n", ((double)(end-start)) / CLOCKS\_PER\_SEC);

printf("insertion char sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

//Selection Sort

//My school NO: 2015556451

int no1[] = { 2, 0, 1, 5, 5, 5, 6, 4, 5, 1 };

//My name and surname

char name1[] = "enver atmaca";

int n1 = sizeof(no1) / sizeof(no1[0]);

int charSize1 = strlen(name1);

selectionSort(no1, n1);

t = clock() - t;

printArray(no1, n1);

printf("selection sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

selectionSortChar(name1, charSize1);

t = clock() - t;

printf("%s\n", name1);

printf("selection char sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

//Bubble Sort

//My school NO: 2015556451

int no2[] = { 2, 0, 1, 5, 5, 5, 6, 4, 5, 1 };

//My name and surname

char name2[] = "enver atmaca";

int n2 = sizeof(no2) / sizeof(no2[0]);

int charSize2 = strlen(name2);

bubbleSort(no2, n2);

t = clock() - t;

printArray(no2, n2);

printf("bubble sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

bubbleSortChar(name2, charSize2);

t = clock() - t;

printf("%s\n", name2);

printf("bubble char sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

//Merge Sort

//My school NO: 2015556451

int no3[] = { 2, 0, 1, 5, 5, 5, 6, 4, 5, 1 };

//My name and surname

char name3[] = "enver atmaca";

int n3 = sizeof(no3) / sizeof(no3[0]);

int charSize3 = strlen(name3);

mergeSort(no3, 0, n3 - 1);

t = clock() - t;

printArray(no3, n3);

printf("merge sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

mergeSortChar(name3, 0, charSize3 - 1);

t = clock() - t;

printf("%s\n", name3);

printf("merge char sort time: %f\n", ((double)t) / CLOCKS\_PER\_SEC);

return 0;

}