#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct node {

int okul\_no;

char isim[50];

char soyad[50];

char dersler[4][50];

struct node \*next;

struct node \*previous;

};

struct node \*head = NULL, \*tail = NULL;

void Ekle(int okul\_no, const char \*isim, const char \*soyad, const char \*ders1, const char \*ders2, const char \*ders3, const char \*ders4) {

struct node \*newNode = (struct node \*)malloc(sizeof(struct node));

newNode->okul\_no = okul\_no;

strcpy(newNode->isim, isim);

strcpy(newNode->soyad, soyad);

strcpy(newNode->dersler[0], ders1);

strcpy(newNode->dersler[1], ders2);

strcpy(newNode->dersler[2], ders3);

strcpy(newNode->dersler[3], ders4);

newNode->next = NULL;

newNode->previous = NULL;

if (head == NULL) {

head = tail = newNode;

} else {

tail->next = newNode;

newNode->previous = tail;

tail = newNode;

}

}

void Goster() {

struct node \*ptr = head;

if (head == NULL) {

printf("Liste bos\n");

return;

}

while (ptr != NULL) {

int veriYapilariDersiAlani = 0;

for (int i = 0; i < 4; i++) {

if (strcmp(ptr->dersler[i], "VeriYapilari") == 0) {

veriYapilariDersiAlani = 1;

break;

}

}

if (veriYapilariDersiAlani) {

printf("Isim: %s\n", ptr->isim);

printf("Soyad: %s\n", ptr->soyad);

printf("Okul No: %d\n", ptr->okul\_no);

printf("Ders 1: %s\n", ptr->dersler[0]);

printf("Ders 2: %s\n", ptr->dersler[1]);

printf("Ders 3: %s\n", ptr->dersler[2]);

printf("Ders 4: %s\n\n", ptr->dersler[3]);

}

ptr = ptr->next;

}

}

// Dizi elemanlarını takas etmek için yardımcı fonksiyon

void Takas(struct node \*a, struct node \*b) {

int temp\_no = a->okul\_no;

char temp\_isim[50];

char temp\_soyad[50];

strcpy(temp\_isim, a->isim);

strcpy(temp\_soyad, a->soyad);

memcpy(a->isim, b->isim, sizeof(b->isim));

memcpy(a->soyad, b->soyad, sizeof(b->soyad));

a->okul\_no = b->okul\_no;

memcpy(b->isim, temp\_isim, sizeof(temp\_isim));

memcpy(b->soyad, temp\_soyad, sizeof(temp\_soyad));

b->okul\_no = temp\_no;

}

// Quick Sort algoritması

void QuickSort(struct node \*low, struct node \*high) {

if (low != NULL && high != NULL && low != high && low != high->next) {

struct node \*pivot = low;

struct node \*left = low;

struct node \*right = high;

while (left != right) {

while (left != right && left->okul\_no <= pivot->okul\_no)

left = left->next;

while (right != NULL && right->okul\_no > pivot->okul\_no)

right = right->previous;

if (left != NULL && right != NULL && left != right)

Takas(left, right);

}

Takas(pivot, right);

QuickSort(low, right->previous);

QuickSort(right->next, high);

}

}

// Kümeleme sıralama algoritması (Okul No'ya göre sıralama)

void KumelemeSiralama() {

struct node \*current = head;

struct node \*index = NULL;

int temp\_no;

char temp\_isim[50];

char temp\_soyad[50];

if (head == NULL) {

printf("Liste bos\n");

return;

} else {

while (current != NULL) {

index = current->next;

while (index != NULL) {

if (current->okul\_no > index->okul\_no) {

temp\_no = current->okul\_no;

strcpy(temp\_isim, current->isim);

strcpy(temp\_soyad, current->soyad);

memcpy(current->isim, index->isim, sizeof(index->isim));

memcpy(current->soyad, index->soyad, sizeof(index->soyad));

current->okul\_no = index->okul\_no;

memcpy(index->isim, temp\_isim, sizeof(temp\_isim));

memcpy(index->soyad, temp\_soyad, sizeof(temp\_soyad));

index->okul\_no = temp\_no;

}

index = index->next;

}

current = current->next;

}

}

}

// İsimleri A'dan Z'ye sıralama (ASCII değerlerine göre)

void ADanZyeSiralama() {

struct node \*current = head;

struct node \*index = NULL;

char temp\_isim[50];

char temp\_soyad[50];

int swapped;

if (head == NULL) {

printf("Liste bos\n");

return;

} else {

do {

swapped = 0;

current = head;

while (current->next != index) {

if (strcmp(current->isim, current->next->isim) > 0) {

strcpy(temp\_isim, current->isim);

strcpy(temp\_soyad, current->soyad);

memcpy(current->isim, current->next->isim, sizeof(current->next->isim));

memcpy(current->soyad, current->next->soyad, sizeof(current->next->soyad));

memcpy(current->next->isim, temp\_isim, sizeof(temp\_isim));

memcpy(current->next->soyad, temp\_soyad, sizeof(temp\_soyad));

swapped = 1;

}

current = current->next;

}

index = current;

} while (swapped);

}

}

int main() {

// 50 tane öğrenci ekleme işlemi

Ekle(1, "Ali", "Yilmaz", "VeriYapilari", "Matematik", "Fizik", "Kimya");

Ekle(2, "Ayse", "Kara", "Matematik", "Biyoloji", "VeriYapilari", "Kimya");

Ekle(3, "Mehmet", "Demir", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(1, "Ali", "Yılmaz", "VeriYapilari", "Matematik", "Fizik", "Kimya");

Ekle(2, "Ayşe", "Kara", "Matematik", "Biyoloji", "VeriYapilari", "Kimya");

Ekle(3, "Mehmet", "Demir", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(4, "Zeynep", "Yıldız", "Matematik", "Fizik", "Biyoloji", "VeriYapilari");

Ekle(5, "Ahmet", "Can", "Fizik", "Kimya", "Matematik", "VeriYapilari");

Ekle(6, "Elif", "Şahin", "Kimya", "VeriYapilari", "Matematik", "Biyoloji");

Ekle(7, "Merve", "Öztürk", "Matematik", "VeriYapilari", "Fizik", "Kimya");

Ekle(8, "Cem", "Kaya", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(9, "Selin", "Aydın", "Biyoloji", "Matematik", "VeriYapilari", "Kimya");

Ekle(10, "Canan", "Çelik", "Fizik", "VeriYapilari", "Matematik", "Kimya");

Ekle(11, "Kemal", "Yılmaz", "Kimya", "Matematik", "Fizik", "VeriYapilari");

Ekle(12, "Ece", "Demir", "Matematik", "VeriYapilari", "Kimya", "Fizik");

Ekle(13, "Mert", "Yıldız", "VeriYapilari", "Fizik", "Kimya", "Matematik");

Ekle(14, "Sevgi", "Aksoy", "Fizik", "Matematik", "VeriYapilari", "Biyoloji");

Ekle(15, "Murat", "Aydın", "Matematik", "Kimya", "VeriYapilari", "Fizik");

Ekle(16, "Ceren", "Kaya", "VeriYapilari", "Matematik", "Biyoloji", "Kimya");

Ekle(17, "Emre", "Şahin", "Kimya", "Fizik", "VeriYapilari", "Matematik");

Ekle(18, "Aylin", "Öztürk", "Biyoloji", "VeriYapilari", "Matematik", "Fizik");

Ekle(19, "Tolga", "Can", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(1, "Ali", "Yılmaz", "VeriYapilari", "Matematik", "Fizik", "Kimya");

Ekle(2, "Ayşe", "Kara", "Matematik", "Biyoloji", "VeriYapilari", "Kimya");

Ekle(3, "Mehmet", "Demir", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(4, "Zeynep", "Yıldız", "Matematik", "Fizik", "Biyoloji", "VeriYapilari");

Ekle(5, "Ahmet", "Can", "Fizik", "Kimya", "Matematik", "VeriYapilari");

Ekle(6, "Elif", "Şahin", "Kimya", "VeriYapilari", "Matematik", "Biyoloji");

Ekle(7, "Merve", "Öztürk", "Matematik", "VeriYapilari", "Fizik", "Kimya");

Ekle(8, "Cem", "Kaya", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(9, "Selin", "Aydın", "Biyoloji", "Matematik", "VeriYapilari", "Kimya");

Ekle(10, "Canan", "Çelik", "Fizik", "VeriYapilari", "Matematik", "Kimya");

Ekle(11, "Kemal", "Yılmaz", "Kimya", "Matematik", "Fizik", "VeriYapilari");

Ekle(12, "Ece", "Demir", "Matematik", "VeriYapilari", "Kimya", "Fizik");

Ekle(13, "Mert", "Yıldız", "VeriYapilari", "Fizik", "Kimya", "Matematik");

Ekle(14, "Sevgi", "Aksoy", "Fizik", "Matematik", "VeriYapilari", "Biyoloji");

Ekle(15, "Murat", "Aydın", "Matematik", "Kimya", "VeriYapilari", "Fizik");

Ekle(16, "Ceren", "Kaya", "VeriYapilari", "Matematik", "Biyoloji", "Kimya");

Ekle(17, "Emre", "Şahin", "Kimya", "Fizik", "VeriYapilari", "Matematik");

Ekle(18, "Aylin", "Öztürk", "Biyoloji", "VeriYapilari", "Matematik", "Fizik");

Ekle(19, "Tolga", "Can", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(1, "Ali", "Yılmaz", "VeriYapilari", "Matematik", "Fizik", "Kimya");

Ekle(2, "Ayşe", "Kara", "Matematik", "Biyoloji", "VeriYapilari", "Kimya");

Ekle(3, "Mehmet", "Demir", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(4, "Zeynep", "Yıldız", "Matematik", "Fizik", "Biyoloji", "VeriYapilari");

Ekle(5, "Ahmet", "Can", "Fizik", "Kimya", "Matematik", "VeriYapilari");

Ekle(6, "Elif", "Şahin", "Kimya", "VeriYapilari", "Matematik", "Biyoloji");

Ekle(7, "Merve", "Öztürk", "Matematik", "VeriYapilari", "Fizik", "Kimya");

Ekle(8, "Cem", "Kaya", "VeriYapilari", "Kimya", "Fizik", "Matematik");

Ekle(9, "Selin", "Aydın", "Biyoloji", "Matematik", "VeriYapilari", "Kimya");

Ekle(10, "Canan", "Çelik", "Fizik", "VeriYapilari", "Matematik", "Kimya");

Ekle(11, "Kemal", "Yılmaz", "Kimya", "Matematik", "Fizik", "VeriYapilari");

Ekle(12, "Ece", "Demir", "Matematik", "VeriYapilari", "Kimya", "Fizik");

Ekle(13, "Mert", "Yıldız", "VeriYapilari", "Fizik", "Kimya", "Matematik");

Ekle(14, "Sevgi", "Aksoy", "Fizik", "Matematik", "VeriYapilari", "Biyoloji");

Ekle(15, "Murat", "Aydın", "Matematik", "Kimya", "VeriYapilari", "Fizik");

Ekle(16, "Ceren", "Kaya", "VeriYapilari", "Matematik", "Biyoloji", "Kimya");

Ekle(17, "Emre", "Şahin", "Kimya", "Fizik", "VeriYapilari", "Matematik");

Ekle(18, "Aylin", "Öztürk", "Biyoloji", "VeriYapilari", "Matematik", "Fizik");

Ekle(19, "Tolga", "Can", "VeriYapilari", "Kimya", "Fizik", "Matematik");

printf("\nGoster : \n");

Goster();

printf("\A-Z : \n");

ADanZyeSiralama();

printf("\Kümeleme : \n");

KumelemeSiralama();

printf("\QuickSort : \n");

QuickSort(head, tail);

return 0;

}

