#include <stdio.h>

#include <string.h>

void add\_employee();

int search\_id( int employee\_id );

void update\_info( int employee\_id );

void delete\_employee( int employee\_id );

double display\_avd\_salary();

void search\_join\_date( int day, int month, int year );

double max\_sal();

int count\_rank( char rank[] );

void search\_name( char name[50] );

struct joining\_date {

int date;

int month;

int year;

};

struct employee\_struct {

int employee\_id;

char employee\_name[50];

char rank[50];

float salary;

char phone\_no[50];

struct joining\_date joining\_date;

};

struct employee\_struct employees[500];

int key = 0;

int main()

{

struct employee\_struct employee;

int option, flag = 1, id, day, month, year, i, rankCount = 0;

double result = 0;

char name[50], rank[50];

while ( 1 ) {

option = menu();

switch ( option ) {

case -1:

flag = 0;

break;

case 1:

add\_employee();

break;

case 2:

printf("Enter your id: ");

scanf("%d", &id);

search\_id( id );

break;

case 3:

printf("Enter your id: ");

scanf("%d", &id);

update\_info( id );

break;

case 4:

printf("Enter your id: ");

scanf("%d", &id);

delete\_employee( id );

break;

case 5:

result = display\_avd\_salary();

printf("Average Salary: %lf\n", result);

break;

case 6:

printf("Enter your date: ");

scanf("%d-%d-%d", &day, &month, &year);

search\_join\_date( day, month, year );

break;

case 7:

result = max\_sal();

for ( i = 0; i <= key; i++ ) {

if ( result == employees[i].salary ) {

printf("%d ", employees[i].employee\_id);

}

}

printf("\n");

break;

case 8:

printf("Enter your Rank:");

getchar();

gets(rank);

rankCount = count\_rank( rank );

printf("The number of employees of this rank is %d\n", rankCount);

break;

case 9:

printf("Enter your name: ");

getchar();

gets(name);

search\_name( name );

break;

}

if ( flag == 0 ) {

break;

}

}

return 0;

}

int menu()

{

int option;

printf("1. Add new employee\n");

printf("2. Search an employee by id\n");

printf("3. Update employee info\n");

printf("4. Delete employee\n");

printf("5. Display average salary\n");

printf("6. Search employee ids after a given joining date\n");

printf("7. Search employee ids having maximum salary\n");

printf("8. Search total no of employees by designation\n");

printf("9. Search employee ids having the given name\n");

printf("Enter -1 to exit.\n");

scanf("%d", &option);

return option;

}

void add\_employee()

{

int i, idFlag = 1, phoneFlag = 1;

struct employee\_struct employee;

printf("Employee ID:");

scanf("%d", &employee.employee\_id);

for ( i = 0; i < key; i++ ) {

if( employees[i].employee\_id == employee.employee\_id ) {

idFlag = 0;

printf("Please provide an unique ID!\n");

return;

}

}

printf("Name of the employee:");

getchar();

gets(employee.employee\_name);

printf("Designation of the employee:");

gets(employee.rank);

printf("Salary of the employee:");

scanf( "%f", &employee.salary );

printf("Phone no of the employee:");

getchar();

gets(employee.phone\_no);

for ( i = 0; i < key; i++ ) {

if( strcmp( employees[i].phone\_no, employee.phone\_no ) == 0 ) {

phoneFlag = 0;

printf("Please provide an unique Phone number!\n");

return;

}

}

printf("Joining date of the employee (dd-mm-yyyy):");

scanf( "%d-%d-%d", &employee.joining\_date.date, &employee.joining\_date.month, &employee.joining\_date.year );

employees[key++] = employee;

}

int search\_id(int id)

{

int i, flag = 0;

for ( i = 0; i <= key; i++ ) {

if ( employees[i].employee\_id == id ) {

flag = 1;

printf("ID: %d\n", employees[i].employee\_id);

printf("Name: %s\n", employees[i].employee\_name);

printf("Designation: %s\n", employees[i].rank);

printf("Salary: %f\n", employees[i].salary);

printf("Phone: %s\n", employees[i].phone\_no);

printf("Joining Date: %d-%d-%d\n\n", employees[i].joining\_date.date, employees[i].joining\_date.month, employees[i].joining\_date.year);

return i;

}

}

if ( flag == 0 ) {

printf("Not Found\n");

return -1;

}

}

void update\_info( int id )

{

struct employee\_struct employee;

int result, i, idFlag = 1, phoneFlag = 1;

result = search\_id( id );

if ( result != -1 ) {

printf("Employee ID:");

scanf("%d", &employee.employee\_id);

for ( i = 0; i < key; i++ ) {

if( employees[i].employee\_id == employee.employee\_id ) {

idFlag = 0;

printf("Please provide an unique ID!\n");

return;

}

}

printf("Name of the employee:");

getchar();

gets(employee.employee\_name);

printf("Designation of the employee:");

gets(employee.rank);

printf("Salary of the employee:");

scanf( "%f", &employee.salary );

printf("Phone no of the employee:");

getchar();

gets(employee.phone\_no);

for ( i = 0; i < key; i++ ) {

if( strcmp( employees[i].phone\_no, employee.phone\_no ) == 0 ) {

phoneFlag = 0;

printf("Please provide an unique Phone number!\n");

return;

}

}

printf("Joining date of the employee (dd-mm-yyyy):");

scanf( "%d-%d-%d", &employee.joining\_date.date, &employee.joining\_date.month, &employee.joining\_date.year );

employees[result] = employee;

printf("Updated Successfully.\n");

}

else {

printf("Not Exists.\n");

}

}

void delete\_employee( int id )

{

struct employee\_struct employee;

int result, i;

result = search\_id( id );

if ( result != -1 ) {

for ( i = result; i < key; i++ ) {

employees[i] = employees[i+1];

}

}

}

double display\_avd\_salary()

{

int i, sum = 0;

double result;

for ( i = 0; i < key; i++ ) {

sum = sum + employees[i].salary;

}

result = sum / i;

return result;

}

void search\_join\_date( int day, int month, int year )

{

int i;

for ( i = 0; i < key; i++ ) {

if ( employees[i].joining\_date.year > year ) {

printf("%d\n", employees[i].employee\_id);

}

else if ( employees[i].joining\_date.year == year ) {

if ( employees[i].joining\_date.month > month ) {

printf("%d\n", employees[i].employee\_id);

}

else if ( employees[i].joining\_date.month == month ) {

if( employees[i].joining\_date.date > day ) {

printf("%d\n", employees[i].employee\_id);

}

}

}

}

}

double max\_sal()

{

double max = employees[0].salary;

int i;

for ( i = 0; i < key; i++ ) {

if ( employees[i].salary > max ) {

max = employees[i].salary;

}

}

return max;

}

int count\_rank( char rank[] )

{

int i, counter = 0;

for ( i = 0; i < key; i++ ) {

if( strcmp( employees[i].rank, rank ) == 0 ) {

counter++;

}

}

return counter;

}

void search\_name( char name[] )

{

int i;

for ( i = 0; i < key; i++ ) {

if( strcmp( employees[i].employee\_name, name ) == 0 ) {

printf("ID: %d\n", employees[i].employee\_id);

}

}

}