#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SIZE 50

typedef struct \_person\* position;

typedef struct \_person {

//data

char fname[32], lname[32];

char birth\_year;

position next;

} person;

position create\_person(position head, char\* fname, char\* lname, int birth\_year)

{

position new\_person = NULL;

//stvaranje prve osobe

new\_person = (position\*)malloc(sizeof(person));

if (!new\_person)

{

printf("error Malloc failed");

return NULL;

}

strcpy(new\_person->fname, fname);

strcpy(new\_person->lname, lname);

new\_person->birth\_year = birth\_year;

new\_person->next = NULL;

}

int prepend\_list(position head, char\* fname, char\* lname, int birth\_year)

{

position new\_person = NULL;

new\_person = create\_person();

new\_person = (position\*)malloc(sizeof(person));

if (!new\_person)

{

printf("error!");

return NULL;

}

new\_person = head->next;

head->next = new\_person;

return 0;

}

int ispis\_osobe(position person)

{

printf("\t%s %s rođen %d. godine\n", person->fname, person->lname, person->birth\_year);

}

int ispis\_liste(position current)

{

while (current != NULL)

{

ispis\_osobe(current);

current = current->next;

}

}

int append\_list(position head)

{

position newPerson = NULL;

newPerson = create\_person();

if (!newPerson)

{

printf("Malloc error");

}

while (head->next!=NULL)

{

head = head->next;

}

newPerson->next = head->next;

head->next = newPerson;

}

int find\_person(position current, char\* sur)

{

while (current != NULL && strcmp(current->lname, sur))

{

current = current->next;

}

return current;

}

int delete\_person(position head, char\* sur)

{

position temp = NULL;

prev = find\_person(head, sur);

if (prev != NULL)

{

temp = prev->next;

prev->next = temp->next;

printf("Osoba izbrisana");

}

else printf("Nije pronađena");

}

int main()

{

char broj = 0;

FILE\* filePointer = NULL;

char lname[32];

person head = {.fname = {0}, .lname = {0}, .bitrth\_year = 0 };

person head = {.fname = {0}, .lname = {0}, .birth\_year = 0 };

printf("1 - dodaj novi element na početak liste\n2- ispiši listu\n3 - dodaj element na kraj liste\n4 - pronađi element\n5- briši element");

do {

printf("\n odaberi broj");

scanf("%c", &broj);

switch (broj)

{

case '1':

{

prepend\_list(&head);

break;

}

case '2':

{

ispis\_liste(head.next);

break;

}

case '3':

{

append\_list(head.next);

break;

}

case '4':

{

scanf("%s", lname);

find\_person(head.next, lname);

break;

}

case '5':

{

scanf("%c", lname);

delete\_person(&head, lname);

break;

}

} while (broj != '0');

}