

1. ZADATAK

Zahtijevane funkcionalnosti implementirati na osnovu komentara prikazanih unutar funkcija. Ukoliko smatrate da je potrebno, program možete proširiti dodatnim funkcionalnostima.

```

//VODITI RAČUNA O POTREBNIM ALOKACIJAMA I DEALOKACIJAMA
enum Oznaka{A, B, C, D, E};
struct Kategorija{
    char datumPolaganja[11];
    Oznaka * _oznakaKategorije;//oznaka će biti pohranjena u dinamičkoj memoriji
    void Unos(char * datum, Oznaka oznaka){
        //Izvršiti potrebne alokacije i odgovarajuću inicijalizaciju atributa strukture }
    void Ispis(){
        /*Ispisati vrijednosti svih atributa strukture */
    }
    void Dealociraj()/*Vrši dealokaciju zauzete memorije*/
};
struct Kandidat{
    char * imePrezime;
    Kategorija * _polozene[5];
    int * _brPolozenih;//brojač položenih kategorija
    void Unos(char * imePrezime){
        //Izvršiti potrebne alokacije i odgovarajuću inicijalizaciju atributa strukture
    }
    void Ispis()/*Ispisati vrijednosti svih atributa strukture */
    bool DodajKategoriju(char * datum, Oznaka * oznaka){
        /*Funkcija omogućava dodavanje podataka o novopoloženoj kategoriji.Korisniku
        onemogućiti da jednom kandidatu doda dvije identične kategorije, te voditi
        računa o mogućim prekoračenjima.U slučaju uspješnog dodavanja kategorije
        funkcija vraća TRUE, a u suprotnom FALSE.*/
    }
    bool UkloniKategoriju(Oznaka * oznaka){
        /*Uklanja kategoriju koja ima identičnu oznaku kao i primljeni parametar. Nakon
        uklanjanja, osigurati kontinuitet elemenata u nizu. U slučaju uspješnog
        uklanjanja kategorije funkcija vraća TRUE, a u suprotnom FALSE.*/
    }
    Kategorija * GetKategoriju(char * datum){
        /*Vraća pokazivač na prvu kategoriju koju je kandidat položio datuma određenog
        parametrom. Ukoliko tražena kategorija ne postoji, funkcija vraća NULL.*/
    }
    void Dealociraj()/*Vrši dealokaciju zauzete memorije*/
};
Kandidat * GetKandidata(Kandidat * niz,int max){
    //Funkcija vraća pokazivač na kandidata koji ima najveći broj kategorija
    //U slučaju da traženi kandidat ne postoji, funkcija vraća NULL
}
void UpisiPodatkeUBinarniFajlu(Kandidat * niz,int max){
    /*Podatke o kandidatima koji su položili određenu kategoriju upisuje u binarni
    fajl.Prilikom upisa, potrebno je očuvati sadržaj koji je do tog momenta
    postojao u fajlu. Oznaku kategorije kao i ime fajla određuje korisnik na
    početku izvršenja funkcije*/
}
int PretragaPoImenuPrezimeni(Kandidat * niz,int max){
    /*Na početku, korisnik unosi dio imena ili prezimena kandidata čije podatke
    želi prikazati. Nakon toga, funkcija vrši pretragu i ispisuje podatke samo onih
    kandidata koju u svom imenu ili prezimenu imaju unesenu riječ (kompletno ime
    ili prezime, jedno ili niz slova, i sl.). Funkcija vraća broj kandidata čije
    su informacije prikazane.*/
}
void main(){
    //Korisniku omogućiti da odredi broj kandidata čije će podatke inicijalizovati
    //Nakon kreiranja liste kandidata demonstrirati korištenje definisanih funkcija
}

```

staticki poz!

app

kandidati

delat---