

PROGRAMIRANJE II (10.02.2021)

- ❶ (15 bodova) Napisati funkciju sa promenljivim brojem argumenata koja formira i vraća dinamički niz realnih brojeva tipa *double*. Funkcija prihvata obavezan parametar *n* koji predstavlja dimenziju niza, a zatim *n* realnih brojeva tipa *double* koji predstavljaju elemente niza. Prototip funkcije je:

```
double* niz_init(int n, ...);
```

Napisati funkciju koja iz ulazne binarne datoteke čiji je naziv parametar *dat* učitava niz realnih brojeva u (dinamičku) memoriju. Ulazna datoteka formatirana je tako da je prvo upisan broj elemenata niza (podatak tipa *int*), nakon čega su upisani elementi niza (podaci tipa *double*). U slučaju da ulazna datoteka ne postoji, funkcija treba da vrati NULL. Preko parametra *n* funkcija treba da vrati dimenziju učitano g niza. Prototip funkcije je:

```
double* niz_citaj(const char *dat, int *n);
```

- ❷ (15 bodova) Neka je dat tip:

```
typedef struct slica {  
    int broj;  
    char naziv[16];  
} SLICICA;
```

kojim se reprezentuju informacije o jednoj samolepljivoj sličici koja se lepi u album sa samolepljivim sličicama.

Napisati funkciju koja niz od *n* sličica sortira po broju sličice u rastućem redosledu korištenjem *insert-sort* algoritma, a čiji je prototip:

```
void sortiraj(SLICICA *niz, int n);
```

Napisati nerekurzivnu funkciju koja vrši interpolaciono pretraživanje (po broju sličice) niza sličica. Funkcija vraća adresu pronađenog podatka ili NULL ako sličica sa datim brojem ne postoji. Prototip funkcije je:

```
SLICICA* trazi(SLICICA *niz, int n, int broj);
```

- ❸ (15 bodova) Neka je dat tip:

```
typedef struct cvor {  
    SLICICA s; // informacioni sadrzaj  
    struct cvor *sledeci;  
} CVOR;
```

kojim se reprezentuje čvor jednostruko povezane liste podataka o samolepljivim sličicama, pri čemu je definicija tipa *SLICICA* data u drugom zadatku.

Napisati funkciju koja dodaje podatke o novoj sličici u uređenu (rastuće po broju sličice) jednostruko povezanu listu. Ako sličica sa datim brojem postoji, ignorisati pokušaj dodavanja sličice. Prototip funkcije je:

```
void dodaj(CVOR **pglava, SLICICA s);
```

Napisati funkciju koja u tekstualnu datoteku upisuje podatke o sličicama (za svaku sličicu podaci se upisuju u zasebnom redu) za koje je ispunjen neki uslov. Prvi parametar funkcije je pokazivač na prvi čvor liste, drugi parametar je pokazivač na tekstualnu datoteku, dok je treći parametar pokazivač na funkciju koja proverava i vraća informaciju o tome da li je neki uslov ispunjen za zadatu sličicu, pri čemu ta funkcija (provera uslova) vraća vrednost 1 ako uslov jeste ispunjen, odnosno vrednost 0 ako uslov nije ispunjen. Prototip funkcije je:

```
void pisi_ako(CVOR *glava, FILE *dat,  
             int (*uslov)(SLICICA));
```

- ❹ (15 bodova) Neka je dat tip:

```
typedef struct cvor {  
    SLICICA s; // informacioni sadrzaj  
    struct cvor *levi, *desni;  
} CVOR;
```

kojim se reprezentuje čvor stabla binarne pretrage podataka o samolepljivim sličicama (broj sličice predstavlja ključ), pri čemu je definicija tipa *SLICICA* data u drugom zadatku.

Napisati rekurzivnu funkciju koja dodaje novi podatak u stablo binarne pretrage. Ako sličica sa datim brojem postoji, ignorisati pokušaj dodavanja sličice. Funkcija treba da vrati koren stabla. Prototip funkcije je:

```
CVOR* dodaj(CVOR *koren, SLICICA s);
```

Napisati rekurzivnu funkciju koja pretražuje stablo po broju sličice. Funkcija treba da vrati adresu pronađenog čvora ili NULL ako sličica sa datim brojem ne postoji. Prototip funkcije je:

```
CVOR* tr_broj(CVOR *koren, int broj);
```

Napisati rekurzivnu funkciju koja pretražuje stablo po nazivu sličice. Funkcija treba da vrati adresu pronađenog čvora ili NULL ako sličica sa datim nazivom ne postoji. Pretpostaviti da su nazivi sličica jedinstveni. Prototip funkcije je:

```
CVOR* tr_naziv(CVOR *koren, const char* naziv);
```