

## PROGRAMIRANJE II (21.06.2016.)

- 1 Napisati funkciju sa promjenljivim brojem argumenata koja čita stringove iz  $n$  tekstualnih datoteka i upisuje u izlaznu tekstualnu datoteku samo one stringove koji ispunjavaju odgovarajući uslov. Funkcija ima sljedeći prototip:

```
void pisi(FILE *out, int (*f)(const char*), int n, ...);
```

Funkcija kao argumente prima izlaznu datoteku u koju se upisuje rezultat, funkciju koja provjerava da li string ispunjava dati uslov, broj ulaznih tekstualnih datoteka  $n$ , te nazive ulaznih tekstualnih datoteka kao neobavezne argumente, respektivno. Smatrati da se u ulaznim tekstualnim datotekama nalaze isključivo stringovi.

- 2 Data je binarna datoteka maksimalne dužine 4GB. Raspoloživa RAM memorija za stek i dinamičku memoriju programa je 2GB. Minimizarati broj čitanja sa diska (odnosno broj poziva funkcija za čitanje). Napisati funkciju koja prebrojava sve bajtove u datoteci koji imaju setovane iste bite kao zbir prva dva bajta datoteke, čiji je prototip:

```
long long count (char* filename);
```

Funkcija treba da prepíše krajnje bajtove datoteke (odgovarajući broj posljednjih bajtova datoteke) sa rezultatom i vrati rezultat. U glavnom programu treba pozvati datu funkciju sa nazivom datoteke koji je prvi argument komandne linije.

- 3 Neka su definisani tipovi:

```
typedef struct stek
{
    void *info;
    struct stek *sljedeci;
} STEK; // ulancana reprezentacija steka
```

```
typedef struct cvor
{
    void *info;
    struct cvor *sljedeci;
} CVOR; // jednostruko povezana lista
```

Napisati funkciju za invertovanje informacionog sadržaja jednostruko povezane liste uz pomoć steka, čiji je prototip:

```
void invertuj(CVOR **pglava);
```

Nije dozvoljeno korišćenje pomoćnih lista tj. mora se invertovati informacioni sadržaj originalne liste.

- 4 Neka su definisani tipovi:

```
typedef struct cvor_s
{
    char c;
    struct cvor_s *lijevi, *desni;
} CVOR_S; // cvor binarnog stabla

typedef struct cvor_r
{
    CVOR_S *cs;
    struct cvor_r *sljedeci;
} CVOR_R; // cvor reda
```

i funkcije za dodavanje i brisanje iz reda:

```
void dodaj(CVOR_R **pf, CVOR_R **pr, CVOR_S *cs);
int brisi(CVOR_R **pf, CVOR_R **pr, CVOR_S **cs);
```

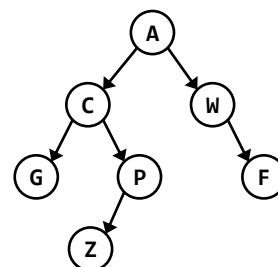
Funkcija *dodaj* omogućava dodavanje adrese čvora binarnog stabla ( $cs$ ) u odgovarajući red. Prvi i drugi argument funkcije predstavljaju adrese odgovarajućih pokazivača na prvi (*front*) i posljednji (*rear*) element reda.

Funkcija *brisi* omogućava dohvatanje (preko parametra  $cs$ ) i brisanje prvog elementa iz reda, a vraća informaciju o tome da li je element uspješno obrisano iz reda (element nije uspješno obrisano iz reda ako je red prazan). Prvi i drugi argument funkcije predstavljaju adrese odgovarajućih pokazivača na prvi (*front*) i posljednji (*rear*) element reda.

Definisati funkciju koja ispisuje sadržaj binarnog stabla ako se stablo obilazi po širini (nivo po nivo, analogno obilasku grafa po širini). Prototip funkcije je:

```
void level_order(CVOR_S *korijen);
```

Npr. za stablo sa slike, rezultat obilaska po širini je: ACWGPfZ.



### Maksimalan broj bodova po zadacima

K1		
1.	2.	Σ
25	25	50

K2		
3.	4.	Σ
25	25	50

Integralno				
1.	2.	3.	4.	Σ
25	25	25	25	100