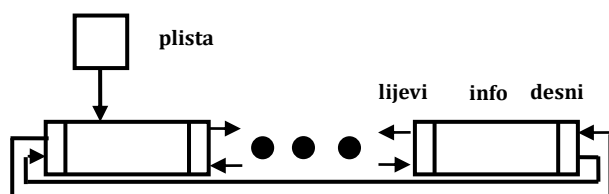


PROGRAMIRANJE II K2 (13.06.2019.) – GRUPA A

- ❶ (5 bodova) Neka je dat sljedeći tip podataka:

```
typedef struct cvor {
    int info; // inf. sadržaj
    struct cvor *lijevi, *desni;
} CVOR;
```

kojim se reprezentuje čvor dvostruko povezane kružne liste i koja se može predstaviti vizuelno na sljedeći način:



Napisati funkciju čiji je prototip:

```
CVOR* dodaj_pocetak(CVOR **plista, int podatak);
```

Funkcija vrši dodavanje novog čvora na početak liste te isti vraća kao rezultat. Pri tome je parametar *plista* spoljašnji pokazivač na listu koji pokazuje na prvi čvor liste, a parametar *podatak* informacioni sadržaj novog čvora liste koji se dodaje na početak iste.

- ❷ (5 bodova) Dat je sljedeći tip podataka koji se koristi pri radu sa ulančanom reprezentacijom reda (FIFO), kao i sa ulančanom reprezentacijom steka (LIFO):

```
typedef struct cvor {
    char c;
    struct cvor *sljedeći;
} CVOR;
```

Pretpostaviti da su implementirane sljedeće funkcije:

```
void dodaj_u_red(CVOR **pf, CVOR **pr, char *c);
int brisi_iz_reda(CVOR **pf, CVOR **pr, char *c);
void brisi_red(CVOR **pf, CVOR **pr);
int ukloni_sa_steka(CVOR **tos, char *c);
void brisi_stek(CVOR **tos);
```

Potrebno je:

- napisati funkciju za dodavanje elementa na stek:
`void dodaj_na_stek(CVOR **tos, char *c);`
- napisati glavnu funkciju (**main**) u kojoj korisnik unosi *n* znakova (mala slova) putem standardnog ulaza (jedan po jedan znak). Prilikom unosa, znakovi se dodaju u red i na stek. Nakon unosa svih znakova potrebno je ispisati riječ koja nastaje od datih znakova i da li je ta riječ palindrom (poređenjem znakova iz reda sa znakovima na steku).

Napomene:

- riječ je palindrom ukoliko se čitanjem u oba smijera zadrži isto značenje (npr. oko, ana, potop, ...),
- funkcije *brisi_iz_reda* i *ukloni_sa_steka* vraćaju vrijednost 1 nakon uspješnog brisanja/uklanjanja elementa iz kolekcije, a u protivnom vrijednost 0 ukoliko je kolekcija bila prazna.

- ❸ (5 bodova) Neka je dat sljedeći tip podataka:

```
typedef struct node {
    char c; // inf. sadržaj
    struct node *left, *right;
} NODE;
```

kojim se reprezentuje čvor stabla binarne pretrage, pri čemu je informacioni sadržaj čvora znak (*c*).

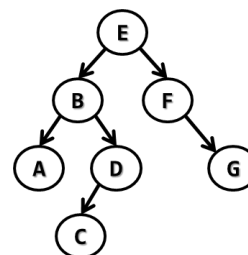
Napisati funkciju koja pretražuje stablo binarne pretrage i vraća informaciju o tome da li se dati znak nalazi u stablu (0 – znak se ne nalazi u stablu, 1 – znak se nalazi u stablu). Prvi parametar funkcije predstavlja korijen stabla (*root*), a drugi ključ za pretragu (*key*). Pored toga, funkcija treba da formira string *str* (treći parametar funkcije) koji se sastoji od znakova koji predstavljaju informacioni sadržaj posjećenih čvorova prilikom pretrage (znakovi treba da budu upisani redoslijedom kojim su čvorovi posječeni). Pretpostaviti da je izvršena odgovarajuća alokacija stringa, kao i inicijalizacija na prazan string, prije poziva funkcije *find*, kao u primjeru. Prototip funkcije je:

```
int find(NODE *root, char key, char str[]);
```

Primjer korištenja funkcije za stablo dato na slici:

```
char s1[10] = "", s2[10] = "";
printf("%d", find(root, 'C', s1)); // 1
printf("%s", s1); // EBDC
printf("%d", find(root, 'Z', s2)); // 0
printf("%s", s2); // EFG
```

Primjer stabla:



- ❹ (5 bodova) Neka su dati tipovi:

```
typedef struct node {
    int ind; // inf. sadržaj
    struct node *next;
} NODE;
```

i

```
typedef struct graph {
    int n; // broj cvorova grafa, 0 <= n <= 50
    char *infos[50]; // inf. sadržaj cvorova grafa
    NODE *heads[50]; // niz zaglavlja
} GRAPH;
```

kojima se reprezentuju čvor jednostruko povezane liste i netežinski graf (ulančana reprezentacija), respektivno.

Napisati funkciju koja određuje i vraća ulazni stepen čvora sa indeksom *ind* datog usmjerenog grafa *g*. Ako je indeks čvora izvan opsega, funkcija treba da vrati vrijednost -1. Prototip funkcije je:

```
int in_degree(const GRAPH *g, int ind);
```