

12. lipnja 2007.

FER2 – ASP – 2006/2007 – GRUPA 9
VJEŽBANJE ZA 3. BLIC

RED-01

Ukoliko je ulaz = 1, a izlaz=4, koliko ima elemenata u redu realiziranom pomoću cirkularnog polja, ako je veličina polja 10 (pretpostavite da ulaz pokazuje na prvi prazan element, dok izlaz pokazuje na prvi stavljeni element)?

- a) 7
 - b) 6
 - c) 3
 - d) 5
 - e) ne može se odrediti
-

RED-02

Neka je na sljedeći način napisana funkcija koja skida element tipa `tip` iz reda realiziranog cikličkim poljem:

```
int SkiniIzReda (tip *element, tip red[], int n,
                int *izlaz, int ulaz) {
    if (ulaz == *izlaz) return 0;
    (*izlaz) ++;
    *izlaz %= n;
    *element = red[*izlaz];
    return 1;
}
```

Koja je od sljedećih tvrdnji **lažna**?

- a) Složenost funkcije je $O(1)$.
 - b) Funkcija vraća 0, ako se iz reda može skinuti točno jedan element.
 - c) Za poziv funkcije, kada u redu postoji barem jedan zapis koji se može skinuti iz reda, funkcija vraća 1.
 - d) Funkcija vraća 1, ako je zapis uspješno skinut iz reda.
 - e) Za poziv funkcije, kada je red prazan, funkcija vraća 0.
-

RED-03

U red realiziran jednostruko povezanom listom pohranjuju se zapisi koji sadrže cijele brojeve. Prototip funkcije za skidanje zapisa iz tako realiziranog reda je (funkcija vraća 1 ili 0, ovisno o tome je li zapis uspješno skinut iz reda):

- a) `int skini (cvor **ulaz, cvor **izlaz, int element);`
 - b) `int skini (cvor **ulaz, cvor **izlaz, int *element);`
 - c) `void skini (cvor **ulaz, cvor **izlaz, int *element);`
 - d) `int skini (cvor *ulaz, cvor **izlaz, int element);`
 - e) `int skini (cvor *ulaz, cvor **izlaz, int *element);`
-

LISTE-01

Predložena je funkcija za pronalazak čvora sa zadanom cjelobrojnomo šifrom u jednostruko povezanoj linearnoj listi:

```
cvor *trazi(cvor *glava, int sifra){
    if (glava->sifra == sifra) return glava;
    if (glava->sljed)
        return trazi(glava->sljed, sifra);
    else
        return NULL;
}
```

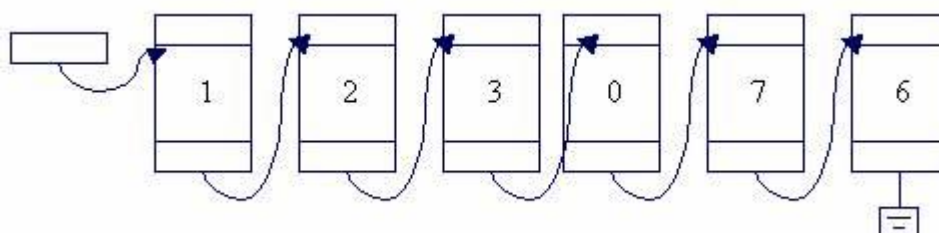
Koja od sljedećih tvrdnji (koje se odnose na predloženu funkciju) je ispravna?

- a) Funkcija je ispravna i vraća pokazivač na čvor sa zadanom šifrom ili NULL ako čvor sa zadanom šifrom ne postoji u listi.
 - b) Funkcija nije ispravna: pretražuje sve čvorove liste osim zadnjeg!
 - c) Funkcija nije ispravna: ne radi za praznu listu (uzrokuje pogrešku kod poziva)!
 - d) Funkcija nije ispravna: ukoliko čvor sa zadanom šifrom ne postoji u listi, funkcija neće "nikad" završiti
 - e) Funkcija nije ispravna: pretražuje sve čvorove liste osim prvog!
-

LISTE-02

Što će vratiti priložena funkcija za zadanu jednostruko povezanu linearnu listu:

Slika (unutar čvorova prikazana je vrijednost varijable `element`):



```

int f(cvor *glava){
    if (glava){
        if (glava->element > 3)
            return glava->element + f(glava->sljed);
        else
            return f(glava->sljed);
    }else{
        return 0;
    }
}

```

- a) 6
- b) 19
- c) 0
- d) 13
- e) 16

LISTE-03

U dvostruko povezanu listu spremaju se zapisi sljedećeg tipa:

```

typedef struct s1{
    int mbr;           // matični broj studenta
    char ime[40+1];    // ime studenta
    float prosjek;     // prosjek ocjena
    struct s1 *sljed;
    struct s1 *preth;
} zapis;

```

Kako glasi funkcija koja izbacuje prvi element iz liste, oslobađa zauzetu memoriju te vraća 1 ako je operacija uspješna, a 0 ako operacija nije uspješna?

a)

```

int izbaci(zapis **glava, zapis **rep) {
    if (*glava == NULL) return 0;
    if ((*glava)->sljed == NULL) {
        free(*glava);
        *glava = *rep = NULL;
    }
    else {
        *glava = (*glava)->sljed;
        free((*glava)->preth);
        (*glava)->preth = NULL;
    }
    return 1;
}

```

b)

```
int izbaci(zapis **glava) {
    if (*glava == NULL) return 0;

    *glava = (*glava)->sljed;
    free((*glava)->preth);
    (*glava)->preth = NULL;
    return 1;
}
```

c)

```
int izbaci(zapis *glava, zapis *rep) {
    if (glava == NULL) return 0;
    if (glava->sljed == NULL) {
        free(glava);
        glava = rep = NULL;
    }
    else {
        glava = glava->sljed;
        free(glava->preth);
        glava->preth = NULL;
    }
    return 1;
}
```

d)

```
int izbaci(zapis **glava) {
    zapis *pom = *glava;
    if (*glava == NULL) return 0;
    glava = (glava)->sljed;
    free(pom);
    return 1;
}
```

e)

```
void izbaci(zapis **glava, zapis **rep) {
    if (*glava != NULL) {
        if ((*glava)->sljed == NULL) {
            free(*glava);
            *glava = *rep = NULL;
        }
        else {
            *glava = (*glava)->sljed;
            free((*glava)->preth);
            (*glava)->preth = NULL;
        }
    }
}
```

STABLA-01

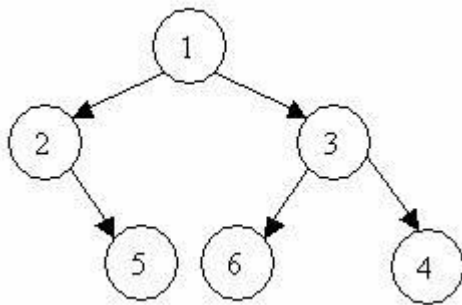
Što će se ispisati funkcijom:

```
void ispis(cvor *korijen){  
    if (korijen) {  
        ispis(korijen->desno);  
        ispis(korijen->lijevo);  
        printf("%c", korijen->element);  
    }  
}
```

za stablo na slici pozivom funkcije

```
ispis (korijen);
```

ako je korijen u trenutku poziva pokazivač na korijen stabla?



- a) 123456
- b) 146352
- c) 643521
- d) 321
- e) 463521

STABLA-02

Koliko razina ima potpuno binarno stablo koje sadrži 100 čvorova i koliki je broj čvorova na posljednjoj razini ?

- a) broj razina=7 broj čvorova=37
- b) broj razina =6 broj čvorova=64
- c) broj razina =7 broj čvorova=64
- d) broj razina =7 broj čvorova=50
- e) broj razina =6 broj čvorova=50

STABLA-03

Što radi slijedeća funkcija?

```
int f(cvor *glava) {  
    int i = 0;  
    if (glava) {  
        if (glava->lijevo || glava->desno) i++;  
        i += f(glava->lijevo);  
        i += f(glava->desno);  
    }  
    return i;  
}
```

- a) Broji čvorove u stablu koji imaju oba djeteta.
 - b) Broji čvorove u stablu koji imaju lijevo dijete.
 - c) Broji čvorove u stablu koji imaju desno dijete.
 - d) Broji čvorove u stablu koji nisu listovi (imaju bar jedno dijete).
 - e) Ništa od navedenog.
-

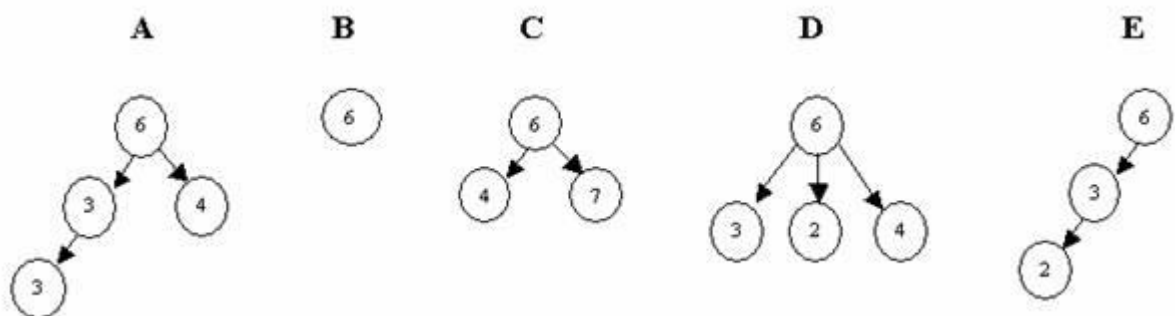
GOMILA-01

Najefikasniji algoritam stvaranja gomile od n elemenata za najgori slučaj ima složenost

- a) $O(1)$
 - b) $O(n)$
 - c) $O(n \cdot \log_2 n)$
 - d) $O(n^2)$
 - e) $O(\log_2 n)$
-

GOMILA-02

Koja od prikazanih stabala su gomile:



- a) A, B
- b) B
- c) A, B, E
- d) A, B, C, E
- e) A

GOMILA-03

Neka se u gomili koja je pohranjena u polje nalaze sljedeći podaci:

66	57	32	30	36	
----	----	----	----	----	--

Što će se desiti kada se u takvu gomilu doda podatak 35 (a nakon što se pozove funkcija *ubaci* koja dodaje elemente u gomilu)?

- a) Izbaci se broj 66, a 35 se umetne na to mjesto
- b) Taj će se broj dodati na kraj polja jer je time očuvano svojstvo gomile
- c) Broj 35 će zamijeniti mjesto s brojem 32
- d) Broj 35 će zamijeniti mjesto s brojem 36
- e) Broj 35 će zamijeniti mjesto s brojem 30

Odgovori

RED-01: a)

RED-02: b)

RED-03: b)

LISTE-01: c)

LISTE-02: d)

LISTE-03: a)

STABLA-01: e)

STABLA-02: a)

STABLA-03: d)

GOMILA-01: b)

GOMILA-02: a)

GOMILA-03: c)