

Drugi Kolokvijum iz Programiranja 2

Kolokvijum traje 90 minuta

Napomene:

- Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
- Vrednost odgovora: tačan = 9; netačan = -2.25; nevažeći (nula ili više zacrnjenih kružića) = 0.
- Na pitanjima se može osvojiti najviše 27 poena.
- Zadatak nosi 23 poena.

I. ZADACI

1) Na programskom jeziku C napisati program za utvrđivanje porekla propuštenih telefonskih poziva. Tekstualna datoteka **Pozivi.txt** sadrži informacije o propuštenim pozivima i sledećeg je formata: svaki red datoteke sadrži podatke o jednom pozivu, a svaki poziv se sastoji od broja telefona (znakovni niz maksimalne dužine 15 karaktera, iza kog sledi razmaknica), i vremena poziva (znakovni niz dužine tačno 5 karaktera). Program treba da poredi brojeve telefona iz ove datoteke sa brojevima telefona koji se nalaze u tekstualnoj datoteci **Imenik.txt** i da napravi novu tekstualnu datoteku **Propusteni.txt** koja osim broja telefona sa kojeg je upućen poziv i vremena poziva sadrži i ime pronađeno u imeniku koje odgovara tom broju telefona. Datoteka **Propusteni.txt** treba da zadrži redosled poziva iz datoteke **Pozivi.txt**. U slučaju da dati broj ne postoji u imeniku, umesto imena u datoteku **Propusteni.txt** treba upisati "NEPOZNAT BROJ". Format datoteke **Imenik.txt** je sledeći: svaki red datoteke sadrži podatke za jedan kontakt, a jedan kontakt se sastoji od broja telefona (znakovni niz maksimalne dužine 15 karaktera, iza kog sledi razmaknica), imena i prezimena kontakta (dva znakovna niza razdvojena razmaknicom, svaki maksimalne dužine 20 karaktera). Poznato je da datoteka **Imenik.txt** može da ima najviše 100 redova. Primer sadržaja navedenih datoteka je dat u sledećoj tabeli.

Datoteka Pozivi.txt	Datoteka Imenik.txt	Datoteka Propusteni.txt
+38164123456 12:15	+38111342648 Peric Pera	+38164123456 12:15 Rakic Raka
+38117987654 13:28	+38112624335 Mikic Mika	+38117987654 13:28 NEPOZNAT BROJ
+38111342648 15:00	+38164123456 Rakic Raka	+38111342648 15:00 Peric Pera
+33145512584 23:58		+33145512584 23:58 NEPOZNAT BROJ

II. PITANJA

1) Koja od sledećih tvrdjenja su tačna za sledeći program na programskom jeziku C?

```
#include <stdio.h>
int ppl(void **a) {
    return **((int **)a)++;
}
main() {
    int s=0, *k=&s;
    char *s1="Jun2007", *s2="Jun2007";
    s = ppl(**k);
    if( s1 == s2 )
        printf("isto\n");
}
```

- U glavnom programu, poziv potprograma `ppl(**k)` nije ispravan.
- Potprogram `ppl` ne menja vrednost formalnog argumenta `a`, već vrednost pokazane lokacije.
- U glavnom programu, vrednost izraza `s1 == s2` je uvek tačna.

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
int rek(int i, int j, int mat[][3]) {
    if( !j ) { printf("%d %d ", i, j); return 1; }
    if( mat[i][j] ) {
        if( rek(i, j-1, mat) || rek(i-1, j, mat) ) {
            printf("%d %d ", i, j);
            return 1;
        }
    }
    return 0;
}
main() {
    int m[][3]={ {1, 1, 0}, {0, 1, 1}, {0, 0, 1} };
    rek(2,2,m);
}
```

A) 11001010

(B) 10111222

C) 10112212

3) Ako su znakovni nizovi ("stringovi") niz1 i niz2 respektivno "ABC" i "DEFG", funkcija uradi vraća pokazivač na znakovni niz:

```
#include <stdlib.h>
#include <string.h>
char *uradi( char *niz1, char *niz2 ){
    char *rez = malloc( 2 * strlen ( niz1 ) + 1 );
    char *pom = rez;
    while ( *pom++ = *niz1++ ) *pom++ = *niz2++ ;
    return rez;
}
```

A) "ABCDEFGF"

B) nula niz

(C) "ADBECF"

4) Dat je deo programskog koda na jeziku C za dodavanje elementa u neopadajuće uređenu jednostruko ulančanu listu, gde se uređivanje vrši po vrednosti polja **br** elemenata liste. Kao argument, funkcija **dođaj** prima pokazivač na element koji treba dodati i pokazivač na prvi element liste, a kao rezultat funkcija vraća pokazivač na prvi element liste nakon dodavanja. Šta treba da stoji na mestu ##### da bi ova funkcija ispravno obavljala tu operaciju?

```
#include <stdlib.h>
#include <stdio.h>
typedef struct elem { int br; struct elem *sl; } Elem;
Elem *dođaj(Elem *e, Elem *prvi){
    Elem *tek=prvi, *pret=NULL;
    if(tek) #####
    else prvi=e,e->sled=NULL;
    return prvi;
}
```

(A)

```
while(tek&&e->br>=tek->br){
    pret=tek;
    tek=tek->sl;
}
if(pret){
    e->sl=pret->sl;
    pret->sl=e;
} else {
    e->sl=prvi,prvi=e;
}
```

B)

```
for(;e->br>tek->br&&tek;){
    e->sl=tek=tek->sl;
    if(pret)
        pret->sl=e;
    pret=tek;
}
```

C)

```
while(tek&&e->br>=tek->br){
    if(e==prvi) prvi=prvi->sl;
    else pret->sl = tek->sl;
    free (e);
    pret=tek;tek=tek->sl;
}
else {
    pret=tek;tek=tek->sl;
}
```