

## Drugi kolokvijum iz Programiranja 2

Kolokvijum traje 90 minuta

### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = **15**; netačan = **-3.75**; nevažeci (nula ili više zacrnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **45** poena.  
 d) Zadatak nosi **55** poena.

### I ZADACI

1) Napisati program na programskom jeziku C koji vrši obradu jednog reda teksta učitano sa standardnog ulaza. Dužina reda nije poznata, tako da je potrebno vršiti dinamičku alokaciju memorije. Na kraju učitavanja rezultujući string treba da zauzima samo onoliko prostora koliko je potrebno za smeštanje neophodnih znakova. Za obradu unetog reda je potrebno implementirati funkciju za podelu sa sledećim prototipom:

```
char** podela(char *str);
```

Funkcija za podelu prihvata niz znakova (string) koji uvek počinje dekadnom cifrom. Dekadna cifra se koristi kao graničnik za podelu stringa na delove (podstringove). Graničnik je uvek jedna dekadna cifra, dok sami podstringovi ne sadrže cifre. Funkcija za podelu treba da vrati niz stringova koji nastaje podelom početnog stringa na osnovu graničnika. Pri tome u niz stringova ulaze samo oni podstringovi čija dužina je manja ili jednaka cifri koja prethodi tom podstringu. Glavni program treba da učitava red teksta sa standardnog ulaza, pozove funkciju za podelu, ispiše dobijeni niz stringova i ponavlja opisani postupak dok se na standardni ulaz ne unese prazan string. Prilikom ispisa rezultujućeg niza stringova, ispisivati jedan string po redu teksta. Voditi računa o ispravnom korišćenju dinamičke memorije.

#### Primer ulaza

```
5alfa3kometa9istina2rad3si
```

#### Primer izlaza

```
alfa
istina
si
```

### II PITANJA

1) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
#include <string.h>
void fja(char **s) {
    while (*s) ++*s++;
}

void main() {
    char *str[4] = { "februar", "mart", "maj", i, s;
    fja(str);
    for (s = i = 0; str[i]; s += strlen(str[i++]));
    printf("%d", s);
}
```

A) 14

(B) 11

C) 13

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
void radi(int *p, int *q) {
    int t;
    while (q - p > 0) {
        t = *p;
        *p++ = *q;
        *q-- = t;
    }
}

void main() {
    int niz[] = { 3, 6, 8, 1, 4, 5, 2 }, n, i;
    n = sizeof(niz) / sizeof(int);
    radi(niz, niz + 2);
    radi(niz + 3, niz + n - 1);
    radi(niz, niz + n - 1);
    for (i = 0; i < n; printf("%d ", niz[i++]));
}
```

(A) 1 4 5 2 3 6 8

B) 8 6 3 3 5 4 1

C) 2 5 4 1 8 6 3

3) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
int meth(int i) {
    return ~(i | i + 1) & ((i | i + 1) + 1);
}

void main() {
    int m[][5] = { { 5, 3, 7 }, { 4, 6 }, { 0, 7, 10, 4 } };
    int *a = *(m+2), *b = *m + 3;
    while (a++, *a)
        *b-- = meth(*a);
    while (*b++)
        printf("%d", *b);
}
```

A) 5340

(B) 24160

C) 2480

4) Koja od sledećih tvrđenja su tačna za programski jezik C?

- A) Ako je definisano `int *p, n=100`; onda pozivi funkcija `p=malloc(n*sizeof(int));` i `p=calloc(n, sizeof(int));` imaju potpuno isti efekat.  
 (B) Ako je pokazivač definisan sa `char (*t) [20] [10]`, rezultat izraza `t + 10` će biti adresa koja je za 2000 veća od adrese koja se nalazi u `t`.  
 C) Pokazivač `p` pokazuje na deo dinamičke memorije memorije veličine `sizeof(int)`. Ako se izvrši naredba `realloc(p, sizeof(int) * 100)`; moguće je da se pokazivaču `p` promenila vrednost.