

**Drugi kolokvijum iz Programiranja 2**Kolokvijum traje **150** minuta**Napomene:**

- a)** Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
**b)** Vrednost odgovora: tačan = **10**; netačan = **-2.5**; nevažeći (nula ili više zacrnjenih kružića) = **0**.  
**c)** Na pitanjima se može osvojiti najviše **50** poena.  
**d)** Zadaci nose po **25** poena.

**I. ZADACI**

**1)** Napisati program na programskom jeziku C za raspoređivanje studenata po salama za polaganje ispita. Na početku programa treba učitati broj sala, broj studenata, i broj mesta u svakoj od sala. Sale treba popunjivati redom počev od onih sa najvećim brojem mesta. Na kraju treba ispisati redne brojeve onih sala koje će se koristiti za polaganje ispita. Maksimalni broj sala i studenata nije unapred poznat, pa se zato mora koristiti dinamička memorija. Po ispisivanju traženih podataka, dealocirati korišćenu dinamičku memoriju.

**2)** Napisati funkciju na programskom jeziku C koji prihvata dva argumenta. Prvi argument je niz stringova koji predstavljaju informacije o ličnim kartama građana. Drugi argument predstavlja broj građana (broj elemenata u prvom argumentu). Svaki string u prvom argumentu se sastoji od nekoliko delova razdvojenih blanko znakom, a to su: najpre reč SUP, potom naziv opštine i na kraju broj lične karte (sedam karaktera). Funkcija treba da odredi i vrati broj građana koji dolaze iz opštine Zvezdara.

**II. PITANJA****1)** Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
main() {
int a[] = { 0, 1, 2, 3, 4 };
int *b[] = { a, a+1, a+2, a+3, a+4 };
int **c = b;
int i = 0;

while( c-b != 5 ) {
if( c-b > 5 ) c -= 5;
**c = *(a + 4 - i);
c += 2; i = (i + 2)%5;
}
for(c--; c>b; c--) printf("%d", **c);
}
```

- (A) 4323                      B) 43243                      C) 43234

**2)** Šta ispisuje sledeća funkcija na programskom jeziku C?

```
#include <stdio.h>
main ( ){
int a[3][3] = { {1,2,3}, {4,5,6}, {7,8,9}}; int i;
for (i=0; i<3; i++) printf("%d %d %d \n", a[2-i][i], *(a[2-i]+1), *((a+i)+2-i)); }
```

- A) 3 2 1                      B) 7 6 1                      C) 7 2 3  
5 5 5                      5 5 5                      5 5 5  
7 8 9                      3 4 9                      3 8 7

**3)** Koja od sledećih tvrdjenja su tačna na programskom jeziku C?

- A) Ako su s1 i s2 pokazivači koji ukazuju na dva znakovna niza (stringa), leksikografski poredak sadržaja ova dva niza možemo utvrditi relacionim operatorima <, > i ==  
**(B)** Ako su promenljive c i t deklarisanе sa char \*c, t[20]; ispravna je dodela vrednosti \*t=\*c;  
C) Ispravna je sledeća deklaracija: char \*((\*f)[](char \*\*));

**4)** Šta ispisuje sledeći program na programskom jeziku C?

```
#include <string.h>
#include <stdio.h>
int strcnt(char *s, int b, char c){
int i, n;
if( b >= strlen(s) ) return 0;
for(i=b, n=0; s[i]; i++)
if( *(s+i) == c ) n++;
return n;
}

void main(){
char *s="Elektrotehnicki fakultet";
int n, b=0;
while(n=strcnt(s,b,'t'))
b+=6, printf("%d", n);
}
```

- (A) 4322                      B) 32100                      C) 32111

**5)** Šta ispisuje sledeća funkcija na programskom jeziku C?

```
void f2(unsigned id) {
if(id > 1 )
{
f2(id/2);
printf("%d", id%2);
}
else
printf("%d", id);
}
```

- A) logaritam sa osnovom 2 broja id  
**(B)** binarni oblik broja id  
C) binarni oblik kvadrata broja id

**6)** Koja od sledećih deklaracija ispravno deklariše niz od 5 pokazivača na funkcije koje vraćaju pokazivače na znakove?

- (A) char \*(\*x[5])()                      B) char \*x[5]()                      C) char \*x()[5]()

**Zadatak 1**

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int br_sala, br_stud;
    int *br_mesta, **sale;
    int i, j;

    printf("Broj studenata:");
    scanf("%d", &br_stud);

    printf("Broj sala: ");
    scanf("%d", &br_sala);

    br_mesta = (int *) malloc(br_sala * sizeof(int));
    sale = (int **) malloc(br_sala * sizeof(int *));

    if( ! br_mesta || ! sale )
    {
        printf("Nedovoljno memorije. Program se prekida.\n");
        exit(0);
    }

    for(i = 0; i < br_sala; sale[i] = br_mesta + i, i++);

    for(i = 0; i < br_sala; i++)
    {
        printf("Kapacitet sale %d : ", i+1);
        scanf("%d", br_mesta + i);
    }

    for(i = 0; i < br_sala-1; i++)
        for(j = i+1; j < br_sala; j++)
            if( *sale[i] < *sale[j] )
            {
                int *t = sale[i];
                sale[i] = sale[j];
                sale[j] = t;
            }

    i = 0;
    while( br_stud > 0 && i < br_sala )
        br_stud -= *sale[i++];

    if( i == br_sala )
        printf("Nema dovoljno sala!\n");
    else
    {
        printf("Rezervisane sale:\n");
        for(j = 0; j < i; j++)
            printf("%d ", sale[j]-br_mesta + 1);
        printf("\n");
    }

    free( sale ); free( br_mesta );
}
```

Napisati program na programskom jeziku C za raspoređivanje studenata po salama za polaganje ispita. Na početku programa treba učitati broj sala, broj studenata, i broj mesta u svakoj od sala. Sale treba popunjavati redom počev od onih sa najvećim brojem mesta. Na kraju treba ispisati redne brojeve onih sala koje će se koristiti za polaganje ispita. Maksimalni broj sala i studenata nije unapred poznat, pa se zato mora koristiti dinamička memorija. Po ispisivanju traženih podataka, deallocirati korišćenu dinamičku memoriju.

**Zadatak 2**

```
#include <string.h>

int nadji_gradjane(char *niz[], int br_elem)
{
    int i, br_gradjana=0;
    char *opstina = "Zvezdara";

    for(i = 0; i < br_elem; i++)
    {
        unsigned int j;

        for(j = 0; j < strlen(opstina); j++)
            if( niz[i][j+4] != opstina[j] )
                break;

        if( j == strlen(opstina) ) br_gradjana++;
    }

    return br_gradjana;
}
```

Napisati funkciju na programskom jeziku C koji prihvata dva argumenta. Prvi argument je niz stringova koji predstavljaju informacije o ličnim kartama građana. Drugi argument predstavlja broj građana (broj elemenata u prvom argumentu). Svaki string u prvom argumentu se sastoji od nekoliko delova razdvojenih blanko znakom, a to su: najpre reč SUP, potom naziv opštine i na kraju broj lične karte (sedam karaktera). Funkcija treba da odredi i vrati broj građana koji dolaze iz opštine Zvezdara.