

**Drugi kolokvijum iz Programiranja 2**Kolokvijum traje **120** 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 koji pronalazi i ispisuje sve one elemente matrice celih brojeva koji u svojoj binarnoj predstavi imaju više aktivnih nego neaktivnih bitova (aktivan bit ima vrednost 1). Smatrati da se ceo broj (`int`) predstavlja na širini od 16 bita. Dimenzije matrice nisu unapred poznate, već se unose sa glavnog ulaza, pa je potrebno koristiti dinamičku alokaciju memorije. Osim dimenzija matrice, sa glavnog ulaza se unose i elementi matrice. Voditi računa o ispravnom dealociranju dinamičke memorije upotrebljene za smeštanje matrice.

**2)** Napisati potprogram na programskom jeziku C koji kao argument prihvata znakovni niz (string). Potprogram određuje i na standardni izlaz ispisuje broj suglasnika, samoglasnika i ostalih karaktera koji se nalaze u sastavu datog znakovnog niza. Sastaviti glavni program koji sa glavnog ulaza učitava 10 reči, od kojih svaka predstavlja jedan string, a zatim za svaku od njih poziva opisan potprogram. Smatrati da reči ne sadrže više od 20 znakova.

**II. PITANJA**

**1)** Koja od sledećih deklaracija ispravno deklarise pokazivač na funkciju koja prihvata argument tipa niz pokazivača na `char`, a vraća podatak tipa pokazivač na `int`?

- A) `int **x(char *y);` B) `int *(*a)(char b[]);` (C) `int *(*p)(char *f[]);`

**2)** Šta treba staviti umesto `####` da bi sledeća funkcija ispravno zamenila sva mala slova velikim u znakovnom nizu koji se dostavlja kao argument?

```
void f(char *z){
    if( z == NULL ) return;
    ####
}
```

- A) `for( ; *z; z++){`  
   `if(*z>='a'&&*z<='z')`  
     `*z+='A'-'a';`  
   `else`  
     `z++;`  
   `}`
- (B) `while(*z){`  
   `if(*z>='a'&&*z<='z')`  
     `*z+='A'-'a';`  
   `z++;`  
   `}`
- C) `do {`  
   `if(*z>='a'&&*z<='z')`  
     `*z+='A'-'a';`  
   `}while(++z);`

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

```
#include <stdio.h>
void main () {
    int n[]={0,5,2,8,3,1,0,6,8};
    int (*b)[2], *p; void *g;

    for(p=n;p<&n[4];++p++){

        g=&n[5]; b=g; g=n;
        for(;b>g;b--) printf("%d ",(*b)[0]);
    }
}
```

- (A) 1 9 6                      B) 1 3 8                      C) 1 3 9

**4)** Šta ispisuje sledeća funkcija na programskom jeziku C, ako se pozove sa `ff(123)` ?

```
void ff(unsigned int broj) {
    unsigned int mask = 0x00A5, change=1;
    while(mask) {
        broj ^= change & mask;
        if(~broj & change) break;
        change<<=2; mask>>=2;
    }
    printf("%d", broj | mask);
}
```

- A) 123                                      (B) 255                                      C) 192

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

- A) Funkcija `malloc` popunjava kompletan alocirani prostor bajtovima vrednosti nula  
 B) Ako je definisano `int **p`, onda `p[1][2]` referiše isti podatak u memoriji kao `*(( *p+1)+2)`  
 (C) Vrednost izraza `sizeof(double*)==sizeof(double)` u opštem slučaju nije jednaka 1

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

```
#include <stdio.h>
int x, y;
void PP(int y, int *z, int a, int b, int *c){
    x=y+1; y=y+x; *z=*z+y; a=a+*z; b=b+a; *c=*c+b;
}
main(){
    x=0; y=1; PP(x,&x,y,y,&y);
    printf("%d %d",y, x);
}
```

- (A) 5 2                                      B) 2 5                                      C) 5 5