

Test pitanja za vežbu u vezi bloka tema 3

1. Označiti tačne iskaze:

- kada bi reč union u deklaraciji unija zamenili sa reč struct dobili bismo deklaraciju strukture
- unije i strukture se po pitanju zauzeća memorije potpuno isto ponašaju
- unije služe da bi se izbeglo dinamičko alociranje memorije
- strukture se koriste radi dinamičkog alociranja memorije

2. Označiti tačan par zauzeća memorije (unija, struktura) u bajtovima ovako deklariranih unija i struktura:

```
typedef union {  
    char a[4];  
    unsigned int b;  
    unsigned long c;  
} unija1;
```

```
typedef struct {  
    char a[4];  
    unsigned int b;  
    unsigned long c;  
} struct1;
```

- a) (4, 10)
- b) (7, 6)
- c) (5, 8)
- d) (9, 7)

3. Pri radu sa kojom vrstom periferija je dozvoljena upotreba RMW komandi?

Ulazna Izlazna Memorijska Ulazno/izlazna

4. Označi tačne iskaze. Atribut `volatile` se koristi ...

- za promenljive koje mogu da promene vrednost na nepredvidljiv način
- za registre periferija
- za promenljive referencirane unutar prekidnih rutina
- za konstante da bi se omogućila njihova promena
- za nizove smeštene u programskoj memoriji

5. Kao alternativa memorijskog mapiranja periferija se koristi emuliranje. Označi sledeće osobine kao tačne ili netačne u odnosu na memorijsko mapiranje:

- emuliranje je brže
- emuliranje traži više signala
- emuliranje uzrokuje veći kod jer je potreban dodatni kod za realizaciju
- emuliranjem se zauzima minimalno potreban broj fizičkih linija

6. Zaokružiti koje su od narednih naredbi RMW tipa.

`x *= y;` `x = y + 1;` `x=x+1; x++;` `x>>=y; x = 1 >> y;`

7. Zaokruži netačne tvrdnje.

- Zauzetost (popunjenost) Rom i Ram memorije zavisi od toga da li programer koristi unije.
- Unija se koristi kao metod za izbegavanje dinamičkog zauzimanja i oslobađanja memorije I ovaj način rada sa memorijom je sličan overlay metodi za lokalne promenljive.
- Unija oslobađa programera od organizacije memorije koju ima na raspolaganju.
- Korišćenje unija smanjuje brzinu rada mikrokontrolera.

8. Zaokružiti tačne iskaze vezane za strukture :

- Način definisanja struktura i unija je različit.
- U strukturi, promena bilo koje promenljive može da utiče na ostale promenljive koje imaju isti memorijski proctor.
- Sve promenljive unutar strukture se preklapaju.

- d) Ukupna veličina strukture odgovara veličini najvećeg elementa.
- e) U strukturi, promena jedne od promenljivih nema nikakvog uticaja na ostale promenljive.

9. Zaokruži jedan od ponuđenih odgovora kako bi rečenica bila tačna:

- a) Ulazno kolo podržava/ne podržava RMW komande.
- b) Promenljive koje imaju odrednicu volatile mogu promeniti svoj sadržaj u bilo kom trenutku/u unapred definisanom trenutku.
- c) U uniji, menjanje bilo koje promenljive može uticati na određene promenljive/uvek utiče na sve promenljive.
- d) C naredba $x \ll= y$ jeste/nije RMW tipa.
- e) Sve promenljive unutar strukture zauzimaju poseban/isti memorijski prostor.

10. Zaokruži tačan odgovor. Koliko bajtova zauzima sledeća unija?

```
typedef union {  
    int x;  
    char y[5];  
    long z;  
} unija1;
```

- a) 4
- b) 5
- c) 7
- d) 11

11. Na primeru unije (oznaci tačno/tačna tvrđenja):

```
typedef union  
{  
    char a;  
    char b[4];  
    int c;  
} data;
```

- a) promenom vrednosti poslednjeg elementa niza b(b[3]), element a se takodje menja
- b) promenom prva dva bajta niza b (b[0] i b[1]) , promenice se vrednost elementa c
- c) moze se menjati svaki element potpuno slobodno, jer to ne utice na ostale elemente
- d) promenom bilo kog bajta elementa b , element a se sigurno nece promeniti

12. Zaokruži pravilan pristup memorijski mapiranoj periferijskoj jedinici čije se vrednosti mogu menjati i u prekidnoj rutini. Pretpostaviti da je veličina eksternog memorijskog prostora 8k.

1. Unsigned char volatile xdata taster 0x8675;
2. Unsigned char volatile xdata motor1 _at_ 15123;
3. Unsigned char xdata motor2 _at_ 0x2423;
4. Unsigned char volatile xdata led _at_ 7234;

13. dopuniti tekstom slike o UI kolima

14. U kojoj liniji se krije greška:

```
#include<stdio.h>
typedef struct Auto{
    char broj_tockova;
    char broj_vrata;
} Automobil;

void init (struct Auto *x){
    x.broj_tockova = 4;
}

int main(void){
    struct Auto mojAuto;
    init(&mojAuto);
    printf("Broj tockova mog auta je: %d", mojAuto.broj_tockova);
    return 0;
}
```

1. struct Auto mojAuto;
2. printf("Broj tockova mog auta je: %d", mojAuto.broj_tockova);
3. x.broj_tockova = 4;
4. init(&mojAuto);

5. void init (struct Auto *x){

15. Šta će biti rezultat nakon kompajliranja AVR-GCC kompajlerom?

```
#include <stdio.h>
int main()
{
typedef union {
    int d;
    char c[3];
} unija;

unija u;

u.c[2]=4;
u.c[1]=5;
u.c[0]=6;
u.d=5;
u.c[2]++;

printf("%d", u.c[2]);
printf("%d", u.c[1]);
printf("%d", u.c[0]);

return 0;
}
```

- a) 156
- b) 505
- c) 105
- d) 556

16. Za svaku od sledećih osobina, napisati kojem načinu povezivanja prilici. (MP-za standardno memorisko mapiranje, EM-za emulaciju potrebnih signala).

- 1) Sve spoljnjše linije koje se koriste za adrese, podatke i kontrolne signale se ne mogu koristiti kao ulazno-izlazni portovi. ____
- 2) Svi signali se generisu softverski. ____
- 3) Periferijsko kolo se spaja kao RAM memorija. ____
- 4) Iz grupe ulazno-izlaznih portova se zauzima minimalan broj fizičkih linija. ____

17. Ukoliko imamo uniju deklarisanu na sledeći način:

```
typedef union{
    unsigned int broj;
    unsigned char niz[2];
} unija;
```

I funkciju definisanu na sledeći način:

```
unsigned char func1(unsigned int a){
    unija un1;
    un1.broj=a;
    un1.broj<<=3;
    return un1.niz[1];
}
```

Koju bajt vrednost ce nam vratiti naredni poziv funkcije ako je način memorisanja višebajtnih promenljivih Big Endian tipa?

```
unsigned int b=0xCA87;
```

```
func1(b);
```

- a) 00111000
- b) 01010100
- c) 10000111
- d) 00011001