

Test pitanja za vežbu u vezi bloka tema 4

1. Koje dva tipa petlji su najbližije:

- a- for i while
- b- for i do-while
- c- while i do-while
- d- sve se uvek različito ponašaju

2. Označi tačne iskaze u vezi break i continue:

- a-break služi i za kompletan prekid izvršavanja petlje
- b-continue služi da se prekine tekuća iteracija petlje i pređe na narednu
- c-u slučaju dvostruke petlje, break osim unutrašnje automatski prekida i spoljašnju
- d-break je obavezno koristiti kod switch (barem jednom)
- e-višestruki continue je bolje koristiti nego break za prekidanje petlji
- f-u slučaju dvostruke petlje, continue pored unutrašnje automatski prekida tekuću interakciju i spoljašnje petlje

3. Označi tačne iskaze u vezi realizacija kašnjenja:

- a- kašnjenja realizovana pomoću petlji su blokirajuća za program
- b- ako se kašnjenja realizuju pomoću petlji, najbolje je koristiti dve parametrizovane petlje
- c- ako je potrebna visoka tačnost kašnjenja najbolje je realizovati u assembleru ili upotrebom simulatora sa merenjem vremena izvršavanja
- d- petlje kašnjenja blokiraju i dozvoljene prekide
- e- za realizaciju kašnjenja preko petlji se preporučuje realizacija dužih kašnjenja
- f- tajmere nije preporučljivo koristiti za realizaciju kašnjenja

4. Označi tačne karakteristike višestruko razgranatih if-else struktura (a la hijerarhija stabla) u odnosu na sekvencijalnu if-else-if strukturu uslovnog izvršavanja:

Dakle

if

if

if

else

else

if

else

else

if

...

u odnosu na:

if

else if

else if

...

a- brže su

b- sporije su

- c- zauzimaju više memorije
- d- zauzimaju manje memorije

5. Označiti tačan rezultat nakon izvršenja sledećeg koda (pažljivo gledajte kod!):

```
#include <stdio.h>

void main(void)
{
    unsigned char n=0;

    if (n==5)
        n=2;
    else
        if (n=2)
        {
            n++;
        }

    switch(n)
    {
        case 1 : printf("%c", 'A');
                break;
        case 2 : printf("%c", 'A');
        case 3 : printf("%c", 'B');
        default: printf("%c", 'C');
    }
}
```

- a) A
- b) ABC
- c) Neće proći prevođenje jer postoji greška
- d) BC

6. Šta sa slike FOR petlje treba izbaciti da bi dobili WHILE petlju:

- a) Treba izbaciti test.
- b) Treba izbaciti samo finalizaciju.
- c) Treba izbaciti samo inicijalizaciju
- d) Treba izbaciti inicijalizaciju i finalizaciju.
- e) Ne treba nista izbaciti, već samo zameniti mesta TRUE i FALSE.

7. Zaokruzi koje od sledećih serijskih komunikacionih protokola koristi mikrokontroler ATmega328.

- a) Ethernet
- b) USART
- c) SATA
- d) CAN
- e) TWI
- f) SPI
- g) PCIe
- h) RS-232

8. Sta ce biti ispisano nakog izvršenja sledeceg koda?

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

int main()
{
    int counter = 0;

    do {
        switch(counter) {
            case 2: printf("%d\n", counter + 1);
            case 4: continue;
            default: printf("%d\n", counter);
        }

        } while (counter++ < 5);

    return 0;
}
```

- a) 0 1 3 3
- b) 0 1 2 3 5
- c) Nece kompajlirati zbog izostanka "break" naredbe iz switch-a.
- d) 0 1 3
- e) 0 1 3 3 5

9. Koja dva obično ne idu zajedno:

- a- Petlja i continue
- b- Switch i continue
- c- Petlja i break
- d- Switch i break

10. Zaokružiti šta će se ispisati pokretanjem sledećeg koda:

```
#include <stdio.h>

int main()
{
    int x = 5;

    for( ; x > 0; --x)
    {
        int x = 6;
        break;
    }
    printf("%d\n", x);
    return 0;
}
```

- a) 4
- b) 5

- c) 6
- d) 0
- e) Neće se ništa ispisati, jer postoji greška u kodu.

11. Zaokruži netacnu tvrdnju:

- a) Ako se iz FOR petlje izostavi testiranje, podrazumeva se tacan rezultat testa.
- b) Posebna primena WHILE petlje je u makro funkcijama, kada je potrebno formirati složenu makro funkciju, sastavljenu od više jednostavnijih konstrukcija.
- c) Osnovna osobina petlji je postojanje skoka unazad.
- d) Mnogi mikrokontroleri imaju masinsku instrukciju DJNZ i ona može da se uklopi u C petlju, narocito DO-WHILE.

12. Šta će biti ispisano prilikom izvršavanja sledećeg koda? (ovo je primer zvani "I'm programmer I have no life" 😊)

```
#include<stdio.h>
int main()
{
    for (int i=1 && printf("i"); i < 3 && printf("t"); ++i &&
printf("f")){
        printf("123");
    }
return 0;
}
```

- a)it123ft123f
- b)it123ft123
- c)itf123tf123
- d)itf123itf