

Nedostaci sljedećih algoritama za MI:

I, J imaju vrijednosti 1 ili 2 (ne istu)

```
uđi_u_KO (I) {  
    J = 3 - I  
    dok je ZASTAVICA != J radi  
        ;  
    ZASTAVICA = I  
}  
izadi_iz_KO (I) {  
    ZASTAVICA = 0  
}
```

Nedostaci sljedećih algoritama za MI:

```
udi_u_KO (I) {  
    J = 1 - I  
    ZASTAVICA[I] = 1  
    PRAVO = J  
    dok je ZASTAVICA[J] == 1 && PRAVO != I radi  
        ;  
}  
  
izadi_iz_KO (I) {  
    ZASTAVICA[I] = 0  
}
```

Nedostaci sljedećih algoritama za MI:

```
udi_u_KO (I) {  
    J = 1 - I  
    ZASTAVICA[I] = 1  
    PRAVO = J  
    dok je ZASTAVICA[J] == 1 && PRAVO == J radi  
        ;  
}  
izadi_iz_KO (I) {  
    ZASTAVICA[I] = 0  
}
```

//ovo je Petersonov algoritam

Nedostaci sljedećih algoritama za MI:

```
udi_u_KO (I) {  
    J = 1 - I  
    ZASTAVICA[I] = 1  
    dok je ZASTAVICA[J] == 1 radi {  
        dok je PRAVO == J radi {  
            ZASTAVICA[I] = 0  
        }  
        ZASTAVICA[I] = 1  
    }  
}  
  
izadi_iz_KO (I) {  
    ZASTAVICA[I] = 0  
    PRAVO = 1 - J  
}
```

Nedostaci sljedećih algoritama za MI:

```
udi_u_KO (I) {  
    J = 1 - I  
    ZASTAVICA[I] = 1  
    dok je ZASTAVICA[J] == 1 radi  
        ZASTAVICA[I] = 0;  
    ZASTAVICA[I] = 1  
}  
izadi_iz_KO (I) {  
    ZASTAVICA[I] = 0  
}
```

Dvije dretve međusobno isključivanje ostvaruju algoritmom:

```
uđi_u_KO() {                                izadi_iz_KO() {
    dok je ( x == 0 )                          x = 1;
        ; //radno čekanje                      }
    x = 0;
}
```

Zašto algoritam nije dobar? Koje kriterije međusobnog isključivanja ne zadovoljava?

Obrazložiti/pokazati na primjeru.

1. Izvođenjem programa s desne strane za konačnu vrijednost varijable a vrijedi (za sve moguće načine izvođenja):

- A) $a == 10$
- B) $a \geq 10$
- C) $a \leq 10$
- D) $a < 10$

```
#include <stdio.h>
#include <pthread.h>
int a = 0, i, t[10];
void *dretva (void *x) {
    a = a + 1; return NULL;
}
int main () {
    for(i=0;i<10;i++)
        pthread_create(&t[i],NULL,dretva, NULL);
    for(i=0;i<10;i++)
        pthread_join(&t[i],NULL);
    printf("%d", a); return 0;
}
```

2. Zajednički nedostatak SVIH algoritama međusobnog isključivanja opisanih u okviru 4. poglavlja (Dekker, Peterson, Lamport, TAS, SWP) je:

- A) ne rade na višeprocorskim sustavima
- B) ne rade na jednoprocesorskim sustavima
- C) trebaju sklopovsku potporu
- D) ne poštuju redoslijed zahtjeva pri propuštanju u kritični odsječak
- E) radno čekanje