

3. Definirati 8086 kod i izgled aktivacionih slogova za funkcije *find* i *count*. Za predstavljanje podataka tipa *int* kao i memorijskih adresa koriste se 2 bajta. Pretpostaviti da se rezultat funkcije smešta u registar CX.

```
struct list
{
    int info;
    struct list* next;
};

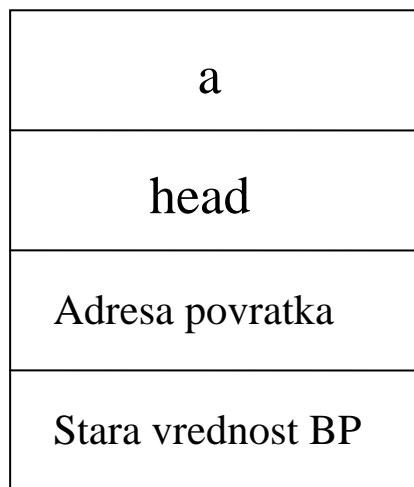
int find(struct list* l, int x)
{
    int res;
    if ( l == 0 )
        res = 0;
    else if ( l->info == x )
        res = 1;
    else
        res = find(l->next, x);
    return res;
}

void count(struct list* head, int a)
{
    static int s;
    s+=find( head, a );
}
```

Rešenje:

Funkcija *count*:

Aktivacioni slog:



8086 kod:

Upis BP na stek

```
PUSH    BP
```

Promena vrednosti pokazivaca BP:

```
MOV     BP, SP
```

Telo funkcije:

```
MOV     AX, [BP+6]
PUSH    AX
MOV     AX, [BP+4]
PUSH    AX,
CALL    find
ADD     SP, 4
MOV     AX, [0100]
ADD     AX, CX
MOV     [0100], AX
```

Uzimasnje stare vrednosti BP-a:

```
POP     BP
RET
```

Funkcija *found*:

Aktivacioni slog:

X
1
Adresa povratka
Stara vrednost BP
res

8086 kod:

upis BP na stek

```
PUSH BP
```

promena vrednosti pokazivaca BP:

```
MOV BP, SP
```

Rezervacija prostora za lokalnu promenljivu:

```
SUB SP, 2
```

Telo funkcije:

```
MOV AX, [BP+4]
```

```
CMP AX, 0
```

```
JNE lab1
```

```
MOV [BP-2], 0
```

```
JMP lab2
```

```
lab1 MOV BX, [BP+4]
```

```
MOV AX, [BX]
```

```
CMP AX, [BP+6]
```

```
JNE lab3
```

```
MOV [BP-2], 1
```

```
JMP lab2
```

Upis stvarnih parametara u stek:

```
lab3 MOV AX, [BP+6]
```

```
PUSH AX
```

```
MOV BX, [BP+4]
```

```
MOV AX, [BX+2]
```

```
PUSH AX
```

Poziv funkcije:

```
CALL find
```

Izbacivanje stvarnih parametara iz steka:

```
ADD SP, 4
```

Smestanje rezultata u CX:

```
lab2 MOV CX, [BP-2]
```

Izbacivanje lokalnih promenljivih iz steka:

```
MOV SP, BP
```

Uzimasnje stare vrednosti BP-a:

```
POP BP
```

```
RET
```