|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vegova Ljubljana | Laboratorijske vaje | Ime in priimek: | Kriterij  < 50% 1 < 65% 2 < 78 % 3 < 90% 4 ≥ 90% 5 | Točke | Ocena |
| Tehniška gimnazija |  |  |  |  |
| 2. letnik | [ A ] | Datum: |  |  |
|  | Razred: | / |  |

**Navodila za reševanje**

Dokument najprej shranite na namizje in **dodajte ime in priimek.** Vse odgovore pišete sem notri. Dovoljena je uporaba dogovorjene literature in dokumentov v tej datoteki, vendar zgolj neposredno kopiranje ne bo dovolj. **Izmenjava zapiskov in** **poraba sredstev za instantno sporočanje ni dovoljena. Naloge lahko izbirate in sicer morate rešiti: dve nalogi za 1 t, eno nalogo za 2 t in vsaj dve nalogi za 3 t. Potrebno je zbrati 10 točk.**

# ZBIRNIK AVR

1. Med seboj zamenjajte vsebino registrov r16 in r17 z uporabo registra r18.

( 1 t )

ldi r16, 0x01

ldi r17, 0x04

mov r18, r16

mov r16, r17

mov r17, r18

1. Kaj je problem, da prevajalnik noče prevesti naslednjega ukaza?

Število 0x0124 je 16 bitno ( 1 t )

ldi r12, 0x0124 ; v r12 naložimo vrednost 0124(16)

1. 8-bitno število (denimo, da je število življenj) na naslovu 0x0200 zmanjšajte za 1.

ldi r16, 0x05

sts 0x0200, r16

lds r17, 0x0200

dec r17

sts 0x0200, r17 ( 1 t )

1. Na primeru prikaži uporabo ukaza **LPM**! Dodaj pomenljive komentarje. ( 2 t )
2. Napišite **podprogram**, ki bo testiral, ali je šestnajstbitno število, ( 2 t )  
   shranjeno v registru Z (r31:r30) enako številu v registru Y (r29:r28). Rezultat naj vrne v zastavici Z.   
   Za demonstracijo najprej napišite začetne nastavitve in nato kličite podprogram!
3. Napišite podprogram, ki bo preveril, ali je 8-bitno število v podatkovnem pomnilniku na naslovu   
   v registru X, enako 0. Če je, vrne v r0 vrednost 1, drugače pa 0 ( 3 t )
4. Napišite zanko, ki bo 10x klicala **podprogram** z imenom *pavza*. ( 3 t )

.org 0x0000 ; Začetni naslov

ldi r16, 10 ; Nastavimo števec na 10

loop:

call pavza ; Kličemo podprogram pavza

dec r16 ; Zmanjšamo števec

brne loop ; Preverimo, ali je števec različen od nič, in se vrnemo v zanko, če je

done:

rjmp done ; Neskončna zanka

.org 0x0100 ; Naslov podprograma pavza

pavza:

ret ; Vrnemo se iz podprograma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vegova Ljubljana | Laboratorijske vaje | Ime in priimek: | Kriterij  < 50% 1 < 65% 2 < 78 % 3 < 90% 4 ≥ 90% 5 | Točke | Ocena |
| Tehniška gimnazija |  |  |  |  |
| 2. letnik | [ B ] | Datum: |  |  |
|  | Razred: | / |  |

**Navodila za reševanje**

Dokument najprej shranite na namizje in **dodajte ime in priimek.** Vse odgovore pišete sem notri. Dovoljena je uporaba dogovorjene literature in dokumentov v tej datoteki, vendar zgolj neposredno kopiranje ne bo dovolj. **Izmenjava zapiskov in** **poraba sredstev za instantno sporočanje ni dovoljena. Naloge lahko izbirate in sicer morate rešiti: dve nalogi za 1 t, eno nalogo za 2 t in vsaj dve nalogi za 3 t. Možno je zbrati 10 točk.**

# ZBIRNIK AVR

1. Med seboj zamenjajte vsebino registrov r18 in r20, tako da uporabite sklad. ( 1 t )

ldi r18, 0x03

ldi r20, 0x04

push r18

push r20

pop r18

pop r20

1. Kaj je problem z naslednjim ukazom?

And se lahko uporablja samo z registri ( 1 t )

and r10, 0x6F ; naredimo operacijo AND med r10 in vrednostjo vrednost 6F(16)

1. Na primeru prikaži uporabo ukaza ST! Dodaj pomenljive komentarje. ( 1 t )
2. Napišite podprogram, ki bo 8-bitno število na naslovu 0x0220 delil s štiri. Ostanek zanemarite. ( 2 t )
3. Napišite **podprogram**, ki bo za en bit v desno premaknil šestnajstbitno število, ( 2 t )  
   shranjeno v registru Z (r31:r30).  
   Za demonstracijo najprej napišite začetne nastavitve in nato kličite podprogram!
4. Napišite podprogram, ki od klica do naslednjega ukaza po povratku potreboval 20 ciklov. ( 3 t )

call u20\_ciklov

ldi r17, 0x03

u20\_ciklov:

call add\_2

add\_2:

mul r16, r17

dec r16

dec r16

ret

1. Napišite **podprogram**, ki bo vsebino podatkovnega pomnilnika   
   od 0x0200 do vključno 0x021F postavil na 0. ( 3 t )