

1. Objasni Funkcionalnost Instrukcija LDA, STA i TAX: Što ove instrukcije rade?
 - a. LDA – upisuje neku vrijednost u akumulator
 - b. STA – sprema trenutnu vrijednost akumulatora na određenu poziciju
 - c. TAX – prebaci trenutnu vrijednost akumulatora u x registar
2. Napiši instrukciju koja postavlja registar A na heksadekadsku vrijednost \$FF.
 - a. LDA #\$FF
3. Napiši kratak program za učitavanje vrijednosti \$07 u registar X, a zatim pohrani ovu vrijednost na lokaciju nulte stranice na \$20.
 - a. LDX #\$07
STX \$0020
4. Napiši dio skupa instrukcija koje se granaju na oznaku SKIP ako je vrijednost u registru A nula.
 - a. BEQ SKIP, CMP #\$00
5. Prikazi kako gurnuti registar A na stog, a zatim ga povući natrag sa stoga.
 - a. PHA – gurne akumulator na stog
 - b. PLA – povuče akumulator sa stoga
6. Napiši jednostavnu podrutinu koja učitava akumulator s vrijednošću \$10 i zatim se vraća iz podrutine.
 - a. JSR Rutina
Rutina:
LDA #\$10
RTS
7. Izvedi bitovni AND između akumulatora koji sadrži \$FF i neposredne vrijednosti \$0F. Koja je rezultat u akumulatoru?
 - a. Neposredna vrijednost \$0F ce se zapisati u akumulator
8. Napiši instrukciju za brisanje zastavice Carry
 - a. CLC
9. Napiši petlju koja smanjuje registar Y s \$05 na \$00, smanjujući za 1 u svakom koraku.
 - a. LDY #\$05
Loop:
DEY
BNE Loop
10. Napiši dio programa koji skoči na adresu \$C000 ako zastavica nula nije postavljena.
 - a. BNE nije_nula
 - b. nije_nula:
 - c. JMP \$C000
11. Usporedi registar X s vrijednošću \$10 i granaj na oznaku EQUAL ako su jednaki.
 - a. LDX #\$10
 - b. CPX #\$10
 - c. BEQ LOOP
 - d. LOOP:
 - e. LDA #\$10
12. Dodaj vrijednost \$01 akumulatoru kada sadrži \$FF. Što se događa sa zastavicom C?
 - a. Zastavica C se mijenja u jedan sto oznacava da postoji carry.

13. Napiši instrukciju za učitavanje akumulatora s vrijednošću pohranjenom na memorijskoj adresi \$0300 i iskoristi offset (pomak) u registru X. Pomak neka bude 5
 - a. LDX #\$05
 - b. LDA #\$01
 - c. STA \$0300
 - d. LDA \$0300, X
14. Prikazi kako koristiti neizravni način adresiranja za učitavanje vrijednosti u registar A.
 - a. LDA #\$00
STA \$10
LDA #\$02
STA \$11
LDY #\$03
LDA (\$10), Y
15. Objasni prednost korištenja adresiranja nulte stranice nad apsolutnim adresiranjem.
 - a. Zero page adresiranje koristi samo \$00 do \$FF, brže je i ne koristi velike byte.
16. Pretvori binarnu vrijednost %10101010 u heksadecimalnu i napiši skup instrukcija za učitavanje ove vrijednosti u akumulator.
 - a. \$AA, LDA #\$AA