

<b>Pradėta</b>	Pirmadienis, 5 gruodžio 2022, 22:47
<b>Būsena</b>	Baigtas
<b>Baigta</b>	Pirmadienis, 5 gruodžio 2022, 23:02
<b>Sugaišta laiko</b>	14 min 41 sek.
<b>Balai</b>	1.00/4.00
<b>Įvertinimas</b>	<b>2.50</b> iš maks. 10.00 ( <b>25%</b> )

## Klausimas 1

Neteisinga

Balas 0.00 iš 1.00

X86 architektūros procesorius laiko registruose žemiau nurodytas reikšmes. Koks bus panaudotas **efektyvus adresas**, vykdant komandą **MOV AX, [BX+0267H]**? Nurodykite atsakymą šešioliktainėje sistemoje, naudojant tiek skaitmenų, kiek reikia pilnam efektyviam adresui užrašyti (jei reikia, su pradiniais nuliais).

A processor with x86 architecture contains values that are given below in its registers. What will be the **effective address** will be used when executing the **MOV AX, [BX+0267H]** instruction? Give the answer in HEX, with just the right number of HEX digits for the full effective address, with leading zeroes if necessary.

AX: 6F7Bh	BP: AF97h	CS: 04E4h	PC: 127Bh
BX: 1891h	SP: ACA8h	SS: A3C2h	
CX: D896h	SI: 710Dh	DS: C25Bh	
DX: A428h	DI: 576Ch	ES: FD4Ah	

Atsakymas: 8A73h



Teisingas atsakymas: 1AF8

## Klausimas 2

Neteisinga

Balas 0.00 iš 1.00

X86 architektūros procesorius laiko registruose žemiau nurodytas reikšmes. Koks bus panaudotas fizinis 20-ies bitų adresas, vykdant komandą **MOV AX, [BX+0267H]**? Nurodykite atsakymą šešioliktainėje sistemoje, naudojant tiek skaitmenų, kiek reikia pilnam 20 bitų adresui užrašyti (jei reikia, su pradiniais nuliais).

A processor with x86 architecture contains values that are given below in its registers. What will be the 20-bit physical address will be used when executing the **MOV AX, [BX+0267H]** instruction? Give the answer in HEX, with just the right number of HEX digits for the full 20 bit address, with leading zeroes if necessary.

AX: 6F7Bh	BP: AF97h	CS: 04E4h	PC: 127Bh
BX: 1891h	SP: ACA8h	SS: A3C2h	
CX: D896h	SI: 710Dh	DS: C25Bh	
DX: A428h	DI: 576Ch	ES: FD4Ah	

Atsakymas: 8A73h



Teisingas atsakymas: 0C40A8

### Klausimas 3

Teisinga

Balas 1.00 iš 1.00

Kokios komandos užkoduotos kiekvienoje žemiau nurodytoje dviejų baitų sekoje? (baitai pateikti šešioliktaine ir dvejetainė sistema)

What instruction is encoded in each two byte sequence below? (bytes are given in HEX and then in binary)

13 F9 00010011 11111001    ADC DI,CX    ✓

88 2C 10001000 00101100    MOV [SI],CH    ✓

03 74 00000011 01110100    ADD SI,[SI+d8]    ✓

Your answer is correct.

Teisingas atsakymas:

13 F9 00010011 11111001 → ADC DI,CX,

88 2C 10001000 00101100 → MOV [SI],CH,

03 74 00000011 01110100 → ADD SI,[SI+d8]

### Klausimas 4

Neteisinga

Balas 0.00 iš 1.00

Kokia reikšmė atsiras išvesties prievade (porte, angl. "output port") mūsų nagrinėtame Harvardo architektūros procesoriuje įvykdžius žemiau nurodytą programą, jei įvesties prievado reikšmė yra **759**? Atsakymą nurodykite kaip skaičių be ženklo dešimtainėje sistemoje.

What value will be output to the output port in the Harvard architecture CPU that was discussed in our course, after executiong the program given below, if the input port has value **759**? Provide the answer as an unsigned decimal number.

```

N:      DS 1                ; The current processed number
CNT:    DS 1                ; Loop invocation counter
        IN 0                ; Input from port 0 to A
        ST N                ; Store the input value into variable N
        LDC 0
        ST CNT              ; Initialise the CNT counter with 0
LOOP:   LD  CNT              ; Increment the counter:
        ADDC 1               ; counter increment is 1
        ST  CNT              ; CNT := CNT + 1
        LD  N                ; Check that N ...
        SUBC 1               ; ... is not 1
        JNZ  CONT            ; When N != 1, perform the loop body
        JMP  OUTPUT          ; When N == 1, finish and output the CNT counter
CONT:   LD  N                ; Check if N is even or odd
        REMC 2
        JNZ  ODD
EVEN:   LD  N                ; If N is even, divide it by 2:
        DIVC 2
        ST  N
        JMP  LOOP
ODD:    LD  N                ; If N is odd, multiply it by 3 ...
        MULC 3
        ADDC 1               ; ... and add 1
        ST  N
        JMP  LOOP
OUTPUT: LD  CNT              ; Output the accumulated counter ...
        OUT 0                ; ... to port 0 (the only port so far ;)
        HALT

```

Atsakymas:

40



Teisingas atsakymas: 60