

## به نام خالق رنگین کمان

### ستاره باباجانی - 99521109

سوال 1: الف) همان طور که در اسلاید ها خواندیم، دستور LDI دیتای 8 بیتی که بعنوان دومین ورودی میگیرد را به general purpose register که رجیستر های شماره 16 تا 31 هستند، میریزد. چون رجیستر دستور ذکر شده در این بازه نیست، این دستور درست کار نمیکند و ارور میدهد.

ب) جواب: گزینه 3

همان طور که در اسلاید ها خواندیم، در ATMEGA32 رجیستر های 8 بیتی وجود دارد. پس بیشترین مقداری که میتوان ذخیره کرد، 11111111 در مبنای دو است. در گزینه ها مبنای هگزادسیمال آن که برابر با 0xFF یا FFH است، وجود دارد.

سوال 2: در این سوال از ما خواسته شده تا کد مربوطه را به زبان اسمبلی ترجمه کنیم. توضیحات مربوط به هر خط کد در جلوی آن آمده است:

```
.INCLUDE "M32DEF.INC"
.DEF cnt = R10           ; Counter variable defined as R10
.EQU i = 0xB0            ; Memory location 0xB0 used for variable i
.EQU j = 0xB1            ; Memory location 0xB1 used for variable j
.ORG 0                   ; Start of the code

    LDI R16, 0           ; Initialize R16 with 0
    MOV cnt, R16         ; Initialize cnt (R10) with 0

    LDI R17, 5           ; Load immediate value 5 to R17
    LDI R18, 6           ; Load immediate value 6 to R18

    LDI R19, 0           ; Initialization for outer loop counter
    LDI R23, 2           ; Value to add to the counter in the outer loop
    LDI R24, 1           ; Value to add to the counter in the inner loop

outerLoop:
    INC R19              ; Increment R19 (outer loop counter)
    STS i, R19           ; Store R19 value into memory location i (0xB0)

    LDI R20, 0           ; Initialize R20
    ADD R20, R17          ; R20 = R17
    ADD R20, R18          ; R20 = R17 + R18 (representing 5+6)

innerLoop:
```

```

innerLoop:
    LDS R21, i          ; Load value from memory location i (0xB0) to R21
    LDS R22, j          ; Load value from memory location j (0xB1) to R22

    CP R22, R21         ; Compare j with i
    ADD cnt, R24        ; Increment cnt
    BRMI innerLoop      ; Branch if j >= i to continue inner loop

    SUB cnt, R24        ; Undo the last increment
    ADD cnt, R23        ; Add 2 to cnt

    CP R19, R20         ; Compare outer loop counter with the total iterations
    BRCS outerLoop      ; Branch if R19 is less than or equal to R20 (total iterations)

```

همان طور که مشاهده میشود، کد درست بوده و اروری ندارد:

ATmega32 memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x000030	48	0	48	32768	0.1%
[.dseg]	0x000060	0x000060	0	0	0	2048	0.0%
[.eseg]	0x000000	0x000000	0	0	0	1024	0.0%

● Assembly complete, 0 errors. 0 warnings

سوال 3: همان طور که سوال گفت، میخواهیم حاصل ضرب دو عدد 16 بیتی را بصورت عددی 32 بیتی محاسبه کنیم. کد زده شده دارای کامنت خط به خط از مراحل انجام آن است:

```

; Define the registers
.DEF R1_HI = R16 ; Define R1_HI register using R16
.DEF R1_LO = R17 ; Define R1_LO register using R17
.DEF R2_HI = R18 ; Define R2_HI register using R18
.DEF R2_LO = R19 ; Define R2_LO register using R19
.DEF RESULT_4 = R20 ; Define RESULT_4 register using R20
.DEF RESULT_3 = R21 ; Define RESULT_3 register using R21
.DEF RESULT_2 = R22 ; Define RESULT_2 register using R22
.DEF RESULT_1 = R23 ; Define RESULT_1 register using R23

; Initialize result registers to 0
LDI RESULT_4, 0 ; Load immediate value 0 into RESULT_4
LDI RESULT_3, 0 ; Load immediate value 0 into RESULT_3
LDI RESULT_2, 0 ; Load immediate value 0 into RESULT_2
LDI RESULT_1, 0 ; Load immediate value 0 into RESULT_1

; Initialize multiplicand and multiplier registers
LDI R1_HI, 0xAB ; Load immediate value 0xAB into R1_HI
LDI R1_LO, 0xCD ; Load immediate value 0xCD into R1_LO
LDI R2_HI, 0xEF ; Load immediate value 0xEF into R2_HI
LDI R2_LO, 0x01 ; Load immediate value 0x01 into R2_LO
LDI R24, 0 ; Load immediate value 0 into R24 (iterator)
LDI R25, 16 ; Load immediate value 16 into R25 (loop counter limit)

; Main Loop

```

```

; Main Loop
MAIN_LOOP:
    LSR R2_HI      ; Shift R2_HI right
    ROR R2_LO      ; Rotate R2_LO right
    BRCC NO_ADDITION ; If Carry Clear, skip addition

    MOV R26, R1_LO ; Move R1_LO to R26
    MOV R27, R1_HI ; Move R1_HI to R27
    LDI R28, 0      ; Load immediate value 0 into R28
    LDI R29, 0      ; Load immediate value 0 into R29
    MOV R30, R24     ; Move R24 to R30 (iterator)

; Loop to calculate the amount to add
CALCULATION_LOOP:
    CP R30, R0       ; Compare R30 with 0
    BREQ ADDITION    ; If R30 is 0, go to ADDITION

    LSL R26          ; Logical Shift Left R26
    ROL R27          ; Rotate Left R27 through Carry
    ROL R28          ; Rotate Left R28 through Carry
    ROL R29          ; Rotate Left R29 through Carry

    DEC R30          ; Decrement R30 (iterator)
    RJMP CALCULATION_LOOP ; Repeat the CALCULATION_LOOP

; Loop for adding to result
; =====

; Loop for adding to result
ADDITION:
    ADD RESULT_1, R26 ; Add R26 to RESULT_1
    ADC RESULT_2, R27 ; Add R27 with carry to RESULT_2
    ADC RESULT_3, R28 ; Add R28 with carry to RESULT_3
    ADC RESULT_4, R29 ; Add R29 with carry to RESULT_4

; Increase the iterator
NO_ADDITION:
    INC R24          ; Increment R24 (iterator)
    CP R24, R25       ; Compare R24 with R25 (loop counter limit)
    BRNE MAIN_LOOP    ; If not equal, repeat MAIN_LOOP

EXIT: RJMP EXIT       ; End of program

```

همان طور که مشاهده میشود، کد درست بوده و اروری ندارد:

Memory use summary [bytes]:							
Segment	Begin	End	Code	Data	Used	Size	Use%
-----							
[.cseg]	0x000000	0x000044	68	0	68	unknown	-
[.dseg]	0x000060	0x000060	0	0	0	unknown	-
[.eseg]	0x000000	0x000000	0	0	0	unknown	-
● Assembly complete, 0 errors. 0 warnings							

سوال 4: همان طور که سوال گفت، میخواهیم حاصل تقسیم دو عدد 16 بیتی را بصورت عددی 16 بیتی محاسبه کنیم. کد زده شده دارای کامنت خط به خط از مراحل انجام آن است:

```
; Define the registers
.DEF R1_HI = R16
.DEF R1_LO = R17
.DEF R2_HI = R18
.DEF R2_LO = R19
.DEF RESULT_2 = R20
.DEF RESULT_1 = R21

; Load initial values into registers
LDI R1_HI, 0x48      ; Load 0x48 into R1_HI
LDI R1_LO, 0x74      ; Load 0x74 into R1_LO
LDI R2_HI, 0x02      ; Load 0x02 into R2_HI
LDI R2_LO, 0x12      ; Load 0x12 into R2_LO
LDI R30, 1           ; Load 1 into R30

; Main Loop

; Main Loop
MAIN:
CP R1_HI, R2_HI      ; Compare R1_HI with R2_HI
BRLO EXIT            ; If R1_HI < R2_HI, exit loop
CP R1_HI, R2_HI      ; Compare R1_HI with R2_HI again
BRNE OPERATION       ; If R1_HI != R2_HI, perform OP
CP R1_LO, R2_LO      ; Compare R1_LO with R2_LO
BRLO EXIT            ; If R1_LO < R2_LO, exit loop

OPERATION:
SUB R1_LO, R2_LO      ; Subtract R2_LO from R1_LO
SBC R1_HI, R2_HI      ; Subtract R2_HI from R1_HI with borrow
ADD RESULT_1, R30      ; Add 1 to RESULT_1
ADC RESULT_2, R0       ; Add carry to RESULT_2
RJMP MAIN             ; Repeat MAIN loop

EXIT: RJMP EXIT        ; End of program
```

همان طور که مشاهده میشود، کد درست بوده و اروری ندارد:

```
Memory use summary [bytes]:
Segment  Begin  End      Code  Data  Used  Size  Use%
-----
[.cseg]  0x000000  0x000022  34    0    34   unknown  -
[.dseg]  0x000060  0x000060   0    0    0   unknown  -
[.eseg]  0x000000  0x000000   0    0    0   unknown  -

Assembly complete, 0 errors. 0 warnings
```