

به نام خالق رنگین کمان

ستاره باباجانی- 99521109

سوال 1:

```
1      call delay
2  delay:                //cycle
3      ldi r20,0x10      // 1
4  L1: ldi r21,0xff      // 1
5  L2: ldi r22,0xff      // 1
6  L3: NOP              // 1
7      dec r22           // 1
8      brne L3           // 2/1
9      dec r21           // 1
10     brne L2           // 2/1
11     dec r20           // 1
12     brne L1           // 2/1
13     ret              // 4
```

همان طور که مشخص است، کد داده شده 3 حلقه تو در تو تا مقادیر 255 و 16 دارد:

$$L1 \rightarrow 1 + 16 * (1 + L2 + 1 + 2) - 1 + 4 = 4173908$$

$$L2 \rightarrow 1 + 255 * (L3 + 1 + 2) - 1 = 260865$$

$$L3 \rightarrow 1 + 255 * (1 + 1 + 2) - 1 = 1020$$

$$So \rightarrow (4173908) * 1\mu s = 4173908 \mu s = 4.173908 s$$

سوال 2: مشکل کد سوال قبل این است که پس از اجرای دستور ret در خط 13، به نقطه call delay باز می‌گردد و این دستور با اینکه stack خالی است دوباره اتفاق می‌افتاد. به منظور جلوگیری از آن، بعد از اجرای ret سیستم باید به label

دیگری پرش کند و جای دستور delay آن را اجرا کند. کد تغییر داده شده به شرح زیر است:

```
1  .include "m32def.inc"
2  start:
3      ldi r16, HIGH(Ramend)
4      out SPH , r16
5      ldi r16, LOW(Ramend)
6      out SPL , r16
7      call delay
8      rjmp another_delay
9
10 delay:
11     ldi r20,0x10
12 L1: ldi r21,0xff
13 L2: ldi r22,0xff
14 L3: NOP
15     dec r22
16     brne L3
17     dec r21
18     brne L2
19     dec r20
20     brne L1
21     ret
22
23 another_delay:
24     rjmp another_delay
```

همان طور که مشاهده میشود بعد از call delay یک jump به label جدید تعریف شده داریم. همچنین stack در ابتدای کد آماده نشده بود که آن نیز درست شد تا خطایی در زمان اجرای ret نباشد.

سوال 3: ابتدا طبق شکل، معادله را حل میکنیم: (سه حلقه تو در تو استفاده شده که به ترتیب 15 و x و 255 در نظر گرفته شده اند.)

```

        ldi r20, 3          // 1
L1: ldi r21,x              // 1
L2: ldi r22,0xff           // 1
L3: NOP                    // 1
        dec r22             // 1
        brne L3             // 2/1
        dec r21             // 1
        brne L2             // 2/1
        dec r20             // 1
        brne L1             // 2/1
        ret                 // 4

```

$$L1: 1 + 3 * (1 + L2 + 1 + 2) - 1 + 4 = 15345x + 64$$

$$L2: 1 + x * (L3 + 2 + 1) - 1 = 1023x$$

$$L3: 1 + 255 * (1 + 1 + 2) - 1 = 1020$$

$$(15345x + 64) * 1\mu s = 1000000 \mu s$$

$$\text{So } \rightarrow x = 65$$

با حل معادله فهمیدیم که حلقه ها به ترتیب باید 15 و 65 و 255 تایی باشند.
حال کد را طبق آن تغییر می‌دهیم: (تاخیر آن حدودا 0.997489 ثانیه است)

```

3  ldi r20,15              ; Load immediate value 15 into register r20 (outer loop counter)
4  L1: ldi r21,65          ; Load immediate value 65 into register r21 (middle loop counter)
5  L2: ldi r22,0xff        ; Load immediate value 0xff into register r22 (inner loop counter)
6  L3: NOP                 ; No operation
7  dec r22                 ; Decrement inner loop counter r22
8  brne L3                 ; Branch if r22 is not equal to zero to label L3 (2 cycles if branch taken, 1 cycle if not)
9
10 dec r21                 ; Decrement middle loop counter r21
11 brne L2                 ; Branch if r21 is not equal to zero to label L2 (2 cycles if branch taken, 1 cycle if not)
12                          ; This loop controls the middle loop, which iterates multiple times
13
14 dec r20                 ; Decrement outer loop counter r20
15 brne L1                 ; Branch if r20 is not equal to zero to label L1 (2 cycles if branch taken, 1 cycle if not)
16                          ; This loop controls the outer loop, which iterates multiple times
17
18 ret                     ; Return from subroutine
19

```

سوال 4: کد برنامه خواسته شده به شرح زیر است:

```

1  ldi  R16 , 0
2  ldi  R17 , 0xff
3  ldi  R18 , 0b11001010
4  out  DDRA , R16 ; port A: input
5  out  DDRB , R16 ; port B: input
6  out  DDRD , R17 ; port D: output
7
8  Q4 : in R19 , PINA
9      cp R18 , R19
10     brne Q4
11     in R20 , PINB
12     add R19 , R20
13     out PORTD , R19
14     inc R18
15     rjmp Q4
16

```

حال سخت افزار طراحی شده به شکل زیر است: (از ATMEGA32 استفاده شده است و پورت های A, B ورودی و پورت D بعنوان خروجی داده شده است.)

