Nom i Cognoms:_____**Una posible solució**_____

1) (3p) Suposant que el clock del sistema és de 8 MHz, calculeu:

a) el temps d'execució del codi en negreta indicat a continuació.
b) amb quins valors caldria inicialitzar COUNT1 i COUNT2 per a que el codi del *Loop* trigui aproximadament 4'5 mil.lisegons.

```
COUNT1 equ 0A ;
COUNT2 equ 0B ;
F equ 1
W equ 0
ACCESS equ 0

...
        movlw 02h
        movwf COUNT1, ACCESS
        movlw 03h
        movwf COUNT2, ACCESS
Loop:   decfsz COUNT1, F, ACCESS
        goto Loop
        movlw 02h
        movwf COUNT1, ACCESS
        decfsz COUNT2, F, ACCESS
        goto Loop
…
```

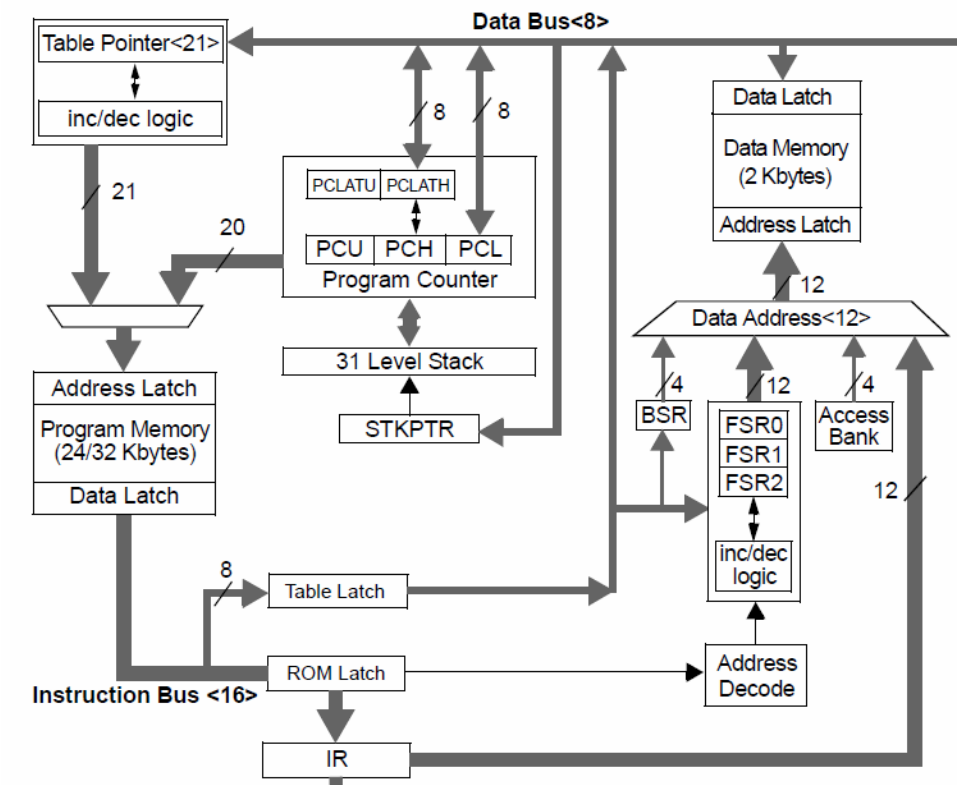
```
...
        movlw 02h
        movwf COUNT1, ACCESS
        movlw 03h
        movwf COUNT2, ACCESS          ;  fins aquí 4 cicles

Loop:   decfsz COUNT1, F, ACCESS
        goto Loop                     ; 3 cicles * COUNT1

        movlw 02h
        movwf COUNT1, ACCESS          ; 2 cicles * COUNT2

        decfsz COUNT2, F, ACCESS
        goto Loop                     ; + 3 cicles * COUNT2
…
```

Ncicles = 4 + [3*COUNT1+5] COUNT2

a) COUNT1 = 2, COUNT2 = 3 donen 37 cicles, és a dir 18,5 microsegons (Tcicle = 0.5 microsegons).

b) Ncicles = 4 + [3*COUNT1+5] COUNT2 = 9000 amb la restricció que COUNT1 i COUNT2 estiguin entre zero i 255. Tenim que [3*COUNT1+5] COUNT2 = 8996, és a dir [3*COUNT1+5] = 8996/COUNT2. Per tanteig, fixem un valor de COUNT2, per exemple a 52, llavors COUNT1 = 56.

2) (1p) En el següent esquema simplificat del PIC18F, a la dreta podem observar un bloc amb la etiqueta **inc/dec logic.** Quin és el significat de la referida etiqueta i quina relació guarda amb l'adreçament a la RAM?



S'està referint al sistema de adreçament indirecte a la memòria RAM amb auto-increment/auto-decrement. Per exemple, qualsevol referència a la adreça especial de la RAM POSTINC2 (@=FDEh) realment provoca un adreçament a l'adreça apuntada pel registre FSR2 i un posterior increment del contingut del registre FSR2.

3) (1p) Si es volgués 'crear' una nova instrucció en llenguatge *assembler* per fer el intercanvi (swap) dels valors que hi han a adreça absoluta F i la adreça F+1:

a) quants bytes ocuparia com a mínim aquesta instrucció?

Degut a que no caldria codificar en el format de la instrucció l'adreça F+1, es podria codificar tot amb 16 bits. Com es habitual en el PIC18F, per codificar una adreça F de 12 bits: 8 bits codificats en la pròpia instrucció i 4 bits en el BSR, més 4 bits per l'OPCODE.

b) en quants cicles s'executaria?

Cal guardar el contingut de l'adreça F en un registre auxiliar [AUX] = [F], posteriorment [F] = [F+1] i per últim [F+1] = [AUX]. Com en un mateix cicle de instrucció es pot fer un adreçament a la memòria RAM per lectura i un adreçament per escriptura, veure per exemple Q Cycle Activity de la instrucció ADDWF en el datasheet, llavors aquesta instrucció es podria implementar en 3 cicles. Cal considerar però, com que hi haurien subcicles improductius possiblement es podria implementar en dos cicles fent algun canvi en l'arquitectura.

Nom i Cognoms:_____**Una posible solució**_____

4) (1'5p.) Com queden els registres de la taula després d'executar el següent codi? Justifiqueu la resposta afegint un comentari després de cada instrucció.

```
movlb  1      ;BSR<–1
movf   0,0,1  ;W<–(0x100)      W=4
movwf  1,0    ;(0x001) <–W     (0x001)= 4
movf   0,0,0  ;W<– (0x000)     W=1
movwf  0,1    ;(0x100) <–W     (0x100)= 1
movf   1,0,0  ;W<– (0x001)     W=4
movwf  0,0    ;(0x000) <–W     (0x000)= 4
```

Bank	Address	Valor previ	Valor final
0	0	1	4
0	1	2	4
0	2	3	3
1	0	4	1
1	1	5	5
1	2	6	6

5) (1'5p.)Detalla tres exemples on la instrucció TSTFSZ provoqui un retard d'un cicle d'instrucció, dos cicles d'instrucció, o tres cicles d'instrucció.

1 CICLE:

```
Tstfsz REG    ; NO es produeix salt
nop
```

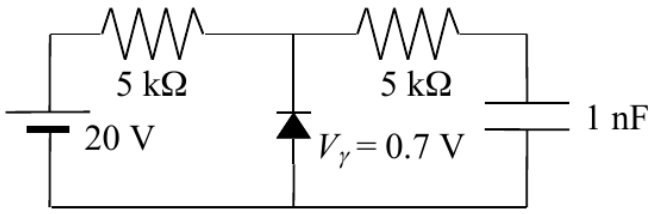
2 CICLES:

```
Tstfsz REG    ; SI es produeix salt
nop
```

3 CICLES:

```
Tstfsz REG    ; SI es produeix salt
Movff A,B     ; instrucció double word
```

6) (2p.) Del circuit de la figura



a) Trobeu la tensió en bornes del condensador un cop assoli el règim estacionari.

El diode està polaritzat inversament, per tant és com si no hi fos.
En règim estacionari el condensador es carrega fins a la tensió màxima.
És a dir, Vc= 20V.

b) i si invertim el diode?

El diode està polaritzat directament, per tant la tensió a l'ànode és de 0.7V.
En règim estacionari el condensador es carrega fins a la tensió màxima.
És a dir, Vc= 0'7V.

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED OPERATIONS									
ADDWF	f, d, a	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1,2
CLRF	f, a	Clear f	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, Skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	4
CPFSGT	f, a	Compare f with WREG, Skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	4
CPFSLT	f, a	Compare f with WREG, Skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1, 2
DECF	f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1, 2
INCF	f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF	f, d, a	Move f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF	f_s, f_d	Move f_s (source) to 1st word f_d (destination) 2nd word	2	1100 1111	ffff ffff	ffff ffff	ffff ffff	None	
MOVWF	f, a	Move WREG to f	1	0110	111a	ffff	ffff	None	
MULWF	f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	None	1, 2
NEGF	f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF	f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	1, 2
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	
RRCF	f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	Set f	1	0110	100a	ffff	ffff	None	1, 2
SUBFWB	f, d, a	Subtract f from WREG with Borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF	f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWFB	f, d, a	Subtract WREG from f with Borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF	f, d, a	Swap Nibbles in f	1	0011	10da	ffff	ffff	None	4
TSTFSZ	f, a	Test f, Skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address