Nom i Cognoms:_____Una possible solució _____

1) Calcula el temps d'execució, en microsegons, del següent tros de codi, suposant que el microcontrolador té un clock de 4MHz. Justifica la resposta.

```
org 0000h   movlw   0x12
            clrf    0x00, 0
            addwf   0x00, 1, 0
            negf    0x00, 0
```

C1	C2	C3	C4	C5
fetch movlw	exe movlw			
	fetch clrf	exe clrf		
		fetch addwf	exe addwf	
			fetch negf	exe negf

5 cicles x 4 subcicles/cicle = 20 subcicles
20 subcicles * 1 subcicle/4 MHz = 5 µs

2) Com queden després de l'execució del codi els següents registres?

	b7	b6	b5	b4	b3	b2	b1	b0
0x00	1	1	1	0	1	1	1	0
W (working register)	0	0	0	1	0	0	1	0

al W queda 0x12 (és hexa, no decimal!) negf: és complement a 2!

3) Considerant que el codi mostrat comença a l'adreça 0x0000, omple el codi màquina corresponent en la següent taula de memòria:

Instruccions:				
movlw 0x12	0000	0001 0010	0001	0000 1110
clrf 0x00, 0	0002	0000 0000	0003	0110 1010
addwf 0x00, 1, 0	0004	0000 0000	0005	0010 0110
negf 0x00, 0	0006	0000 0000	0007	0110 1100

ADDWF **ADD W to f**

Syntax:	ADDWF f {,d {,a}}
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	(W) + (f) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	0010 | 01da | ffff | ffff
Description:	Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

CLRF **Clear f**

Syntax:	CLRF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	000h → f, 1 → Z
Status Affected:	Z
Encoding:	0110 | 101a | ffff | ffff
Description:	Clears the contents of the specified register. If 'a' is '0', the Access Bank is selected.

NEGF **Negate f**

Syntax:	NEGF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	$(\bar{f}) + 1 \rightarrow f$
Status Affected:	N, OV, C, DC, Z
Encoding:	0110 | 110a | ffff | ffff
Description:	Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'. If 'a' is '0', the Access Bank is selected.

MOVLW **Move Literal to W**

Syntax:	MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W
Status Affected:	None
Encoding:	0000 | 1110 | kkkk | kkkk
Description:	The eight-bit literal 'k' is loaded into W.
Words:	1
Cycles:	1

4) Què fa el següent codi?

movlb 2
movf 0x13, 0,1
movlb 3
movwf 0x12,1

Copia el contingut del registre 0x13 del bank 2, al registre 0x12 del bank 3.

5) Podríem fer el mateix que en el codi anterior en una única instrucció? Si és possible, escriu-la:

Si: *movff 0x213,0x312*

6) Compara la sol.lució de la pregunta 4 amb la de la pregunta 5 (si és que n'has trobat alguna).

La de la pregunta 4 ocupa 4 words de 16 bits a memòria de codi, la de la 5, 2 words.
La de la pregunta 4 necessita 4 cicles d'execució, la de la 5, 2 cicles.
La de la pregunta 4 afecta el contingut del Wreg i del BSR.

MOVF — **Move f**

Syntax:	MOVF f {,d {,a}}
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	f → dest
Status Affected:	N, Z
Encoding:	0101 | 00da | ffff | ffff
Description:	The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

MOVWF — **Move W to f**

Syntax:	MOVWF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	(W) → f
Status Affected:	None
Encoding:	0110 | 111a | ffff | ffff
Description:	Move data from W to register 'f'. Location 'f' can be anywhere in the 256-byte bank. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See

MOVFF — **Move f to f**

Syntax:	MOVFF f_s,f_d
Operands:	0 ≤ f_s ≤ 4095 0 ≤ f_d ≤ 4095
Operation:	(f_s) → f_d
Status Affected:	None
Encoding: 1st word (source) 2nd word (destin.)	1100 | ffff | ffff | $ffff_s$ 1111 | ffff | ffff | $ffff_d$
Description:	The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh. Either source or destination can be W (a useful special situation). MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

MOVLB — **Move Literal to Low Nibble in BSR**

Syntax:	MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → BSR
Status Affected:	None
Encoding:	0000 | 0001 | kkkk | kkkk
Description:	The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0' regardless of the value of k_7:k_4.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

Nom i Cognoms:_____Una possible solució _____

7) De quina mida és el bus d'adreces i dades per accedir a la ROM Program Memory.

El bus d'adreces és de 21 bits i el de dades de 16.

8) Es cert o fals que no hi ha cap mena de connexió entre el bus de dades de la ROM i el bus de dades de la RAM? Penseu si hi ha alguna instrucció que ho permeti. Raoneu la resposta.

Tot i que és tracta d'una arquitectura Harvard, amb un bus per accedir al programa i un altre per accedir a les dades, hi han instruccions que mouen dades (constants) des del programa a la memòria de dades. Un exemple d'aquestes instruccions és movlw que mou un literal al working register via el bus de dades (veure diapositiva 3 del PIC-1.PPT).

9) Si el temps d'execució d'un instrucció d'un cicle és de 200 nseg, quina és la freqüència del rellotge? Indiqueu els càlculs en la resposta.

Tclk = Tinstruccio / 4 = 200/4 nseg = 50 nseg
fclk = 1/Tclk = 20 MHz

10) Quina decisió de disseny sobre el format o mida del conjunt d'instruccions obliga a dividir la memòria en bancs?

El fet que la majoria d'instruccions són de 16bits, reservant només 8 bit per a especificar l'adreça a les dades. Com la memòria RAM és més gran que el 256 Bytes, aquesta s'ha dividit en bancs.

11) Si el PIC18F4550 disposa de 2KB de memòria de dades, quants bits del BSR (Bank Select Register) són significatius? I si el microcomputador disposes de 4KB de memòria de dades?

Per adreçar 2KB es necessita 11 bits, dels quals 8 estan especificats en la instrucció i els altres 3 en el registre BSR (Bank Select Register). En el cas de 4KB necessitaríem 4 bits addicionals.

12) Quina instrucció o tipus d'instruccions sol haver en l'adreça 0x00000? Raoneu la resposta.

En l'adreça absoluta 0 de la memòria del programa hi ha la primera instrucció que s'executarà després de donar tensió al microprocessador. Degut a que en adreces posteriors (0x08h i 0x16) hi ha l'espai reservat per a interrupcions, s'acostuma a incloure un salt en l'adreça zero per accedir al programa principal (una instrucció goto o branch).