Nom i Cognoms:_____

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
```


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

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

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	
clrf 0x00, 0	0002		0003	
addwf 0x00, 1, 0	0004		0005	
negf 0x00, 0	0006		0007	

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

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

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

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:_____

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

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.

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.

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

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