ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ Η/Υ

4ο Εξάμηνο

Μαδεμλής Ιωάννης

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 3

Η ΠΡΑΞΗ ΤΗΣ ΑΦΑΙΡΕΣΗΣ

Πράξη	Επεξήγηση
1 1 0 1 -0 1 1 0 1	Μηδέν από ένα μας δίνει 1. Δεν χρειάστηκε δανεικό (δανεικό=0).
1 1 0 1 -0 1 1 0 1Δ **1** 1	Ένα από Μηδέν δεν αφαιρείται. Έτσι παίρνουμε δανεικό και λέμε Ένα (1_2) από Δύο (10_2) μας δίνει 1, άρα γράφουμε ως αποτέλεσμα το 1 και έχουμε 1 δανεικό
1 1 0 1 0 1 1 0 1Δ 1 1 1	Ένα το δανεικό και Ένα μας κάνουν Δύο (10_2), από Ένα (1_2) δεν αφαιρείται. Έτσι παίρνουμε δανεικό και λέμε Δύο (10_2) από Τρία (11_2) μας δίνουν Ένα, άρα γράφουμε ως αποτέλεσμα το 1 και έχουμε 1 ως δανεικό.
1 1 0 1 0 1 1 0 0 1 1 1	Ένα το δανεικό και 0 μας κάνουν ένα (1_2), από Ένα μας δίνει 0 άρα γράφουμε ως αποτέλεσμα το 0 και δεν έχουμε δανεικό. Η πράξη τελείωσε και το αποτέλεσμα είναι $1\,1\,1_2 = 7_{10}$

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

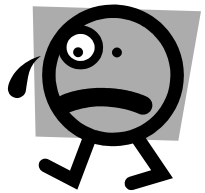
ΕΝΤΟΛΕΣ ΑΦΑΙΡΕΣΗΣ

Η εντολή SUB και οι τρόποι σύνταξής της

Αφαίρεση χωρίς δανεικό, SUB a,b → a=a-b

- SUB καταχωρητής1, καταχωρητής2 — SUB AX,BX
- SUB καταχωρητής, [θέση μνήμης] — SUB AX,[200]
- SUB [θέση μνήμης], καταχωρητής — SUB [200],CX
- SUB καταχωρητής, τιμή — SUB DX, 1AF4
- SUB BY[διεύθυνση μνήμης], τιμή 8bit — SUB BY[200], 7A
- SUB WO[διεύθυνση μνήμης], τιμή 16bit — SUB WO[200], 10F5

~~SUB [θέση μνήμης], [θέση μνήμης]~~



ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΕΣ ΑΦΑΙΡΕΣΗΣ

■ Η εντολή SBB και οι τρόποι σύνταξής της

Αφαίρεση με δανεικό, SBB a,b → a=a-b-Carry

- SBB καταχωρητής1, καταχωρητής2 — SBB AX,BX
- SBB καταχωρητής, [θέση μνήμης] — SBB AX,[200]
- SBB [θέση μνήμης], καταχωρητής — SBB [200],CX
- SBB καταχωρητής, τιμή — SBB DX, 1AF4
- SBB BY[διεύθυνση μνήμης], τιμή 8bit — SBB BY[200], 7A
- SBB WO[διεύθυνση μνήμης], τιμή 16bit — SBB WO[200], 10F5

■ Η εντολή NEG καταχωρητής αλλάζει πρόσημο στην τιμή του καταχωρητή με βάση το συμπλήρωμα ως προς 2

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 3

ΑΣΚΗΣΗ 3.2

Γράψτε ένα πρόγραμμα που να αφαιρεί δύο προσημασμένους αριθμούς των 8 bit που βρίσκονται στις θέσεις μνήμης **200** και **201** και να τοποθετεί το αποτέλεσμα στην θέση μνήμης **202**. Κάντε δοκιμή για τα παρακάτω δεδομένα :

0200		**0201**		**0202**				
16-δικός	10-δικός	16-δικός	10-δικός	Θεωρ. 10-δικός	Εργ. 16-δικός	Εργ. 10-δικός	Σωστό? (Ναι/Οχι)	C/OV/SIGN
6F	**+111**	**70**	**+112**	**-1**	**FF**	**-1**	**NAI**	**CY/NV/NG**
FF	**-1**	**02**	**+2**	**-3**	**FD**	**-3**	**NAI**	**NC/NV/NG**
80	**-128**	**01**	**+1**	**-129**	**7F**	**+127**	**OXI**	**NC/OV/PL**

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 3

ΑΣΚΗΣΗ 3.3

Γράψτε ένα πρόγραμμα που να αφαιρεί τον 16-bit προσημασμένο αριθμό που βρίσκεται στις θέσεις μνήμης **0202**, **0203** (Υ.Τ.Β. στην 0203) από τον 16-bit προσημασμένο αριθμό που βρίσκεται στις θέσεις μνήμης **0200**, **0201**, (Υ.Τ.Β. στην 0201). Βάλτε το αποτέλεσμα της αφαίρεσης στις θέσεις **0204**, **0205,** και το αρνητικό του στις θέσεις μνήμης **0206,0207**.

Κάντε δοκιμή για τα παρακάτω δεδομένα :

Μειωτέος		**Αφαιρέτης**		**Διαφορά**		**-Διαφορά**		**FLAGS (πριν το NEG)**			
200	**201**	**202**	**203**	**204**	**205**	**206**	**207**	**C**	**OV**	**SIGN**	**Σωστό ?**
21	**43**	**10**	**32**	**11**	**11**	**EF**	**EE**	**NC**	**NV**	**PL**	**ΝΑΙ**
CD	**AB**	**11**	**11**	**BC**	**9A**	**44**	**65**	**NC**	**NV**	**NG**	**ΝΑΙ**
00	**80**	**02**	**00**	**FE**	**7F**	**02**	**80**	**NC**	**OV**	**PL**	**ΟΧΙ**

4321	ABCD	8000
-3210	-1 1 1 1	-0002
1 1 1 1	**9 A B C**	**7 F F E**

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

Η ΠΡΑΞΗ ΤΟΥ ΠΟΛ/ΣΜΟΥ ΣΤΟΝ 8088

Πολ/σμός 8 bit: AL * { AL AH, BL BH, CL CH, DL DH } → AX

Πολ/σμός 16 bit: AX * { AX BP, BX SI, CX DI, DX SP } → DX:AX

Το αποτέλεσμα του πολ/σμού είναι σωστό ΠΑΝΤΑ!

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΕΣ ΠΟΛ/ΣΜΟΥ

Η εντολή MUL

Πολ/σμός μη προσημασμένων αριθμών

- MUL καταχωρητής
  - καταχωρητής 8 bit → AX=AL * καταχωρητής
  - καταχωρητής 16 bit → DX:AX=AX * καταχωρητής

Η εντολή IMUL

Πολ/σμός προσημασμένων αριθμών

- IMUL καταχωρητής
  - καταχωρητής 8 bit → AX=AL * καταχωρητής
  - καταχωρητής 16 bit → DX:AX=AX * καταχωρητής

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΑΣΚΗΣΗ 3.4

Γράψτε ένα πρόγραμμα που να πολλαπλασιάζει τους μη προσημασμένους 16 bit αριθμούς που βρίσκονται στις θέσεις μνήμης 0200-0201 και 0202-0203 και να βάζει το αποτέλεσμα στις θέσεις μνήμης 0204-0207.

Εκτελέστε το πρόγραμμά σας για τα δεδομένα του παρακάτω πίνακα:

Πολ/στής 0200...0201	**Πολ/στής 0202...0203**	**Γινόμενο 0204...0207**	**Σωστό ? (Ναι/Οχι)**
34 12	**00 01**	**00 34 12 00**	**N A I**
FF FF	**FF FF**	**01 00 FE FF**	**N A I**

1234	FFFF
x 0100	x FFFF
00123400	**FFFE0001**

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

Η ΠΡΑΞΗ ΤΗΣ ΔΙΑΙΡΕΣΗΣ ΣΤΟΝ 8088

Διαίρεση 16 bit: AX / { AL AH, BL BH, CL CH, DL DH } → Π → AL, Υ → AH

Διαίρεση 32 bit: DX:AX / { AX BP, BX SI, CX DI, DX SP } → Π → AX, Υ → DX

Το αποτέλεσμα της διαίρεσης **ΔΕΝ** είναι πάντα σωστό!

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΕΣ ΔΙΑΙΡΕΣΗΣ

Η εντολή DIV

Διαίρεση μη προσημασμένων αριθμών

- DIV καταχωρητής
  - καταχωρητής 8 bit → AX/καταχωρητής και Π → AL & Y → AH
  - καταχωρητής 16 bit → DX:AX/καταχωρητής και Π → AX & Y → DX

Η εντολή IDIV

Διαίρεση προσημασμένων αριθμών

- IDIV καταχωρητής
  - καταχωρητής 8 bit → AX/καταχωρητής και Π → AL & Y → AH
  - καταχωρητής 16 bit → DX:AX/καταχωρητής και Π → AX & Y → DX

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΑΣΚΗΣΗ 3.5

Γράψτε ένα πρόγραμμα που να διαιρεί τον 32 bit προσημασμένο αριθμό που βρίσκεται στις θέσεις μνήμης 0200-0203, με τον 16-bit προσημασμένο αριθμό που βρίσκεται στις θέσεις μνήμης 0204-0205 και να βάζει το Πηλίκο στις θέσεις μνήμης 0206-0207 και το Υπόλοιπο στις θέσεις μνήμης 0208-0209.

Εκτελέστε το πρόγραμμά σας για τα δεδομένα του παρακάτω πίνακα:

Διαιρετέος **200 201 202 203**	**Διαιρέτης** **204 205**	**Πηλίκο** **206 207**	**Υπόλοιπο** **208 209**
01 40 23 01	**00 10**	**34 12**	**01 00**
00 00 FF FF	**00 80**	**02 00**	**00 00**

01234001/1000 → **Π=1234**, **Υ=0001**

FFFF0000/8000 → **Π=0002**, **Υ=0000**

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΣΥΝΘΗΚΗ ΣΤΗΝ **ASSEMBLY**

Ανώτερες γλώσσες προγραμματισμού	IF συνθήκη=True THEN GOTO διεύθυνση
Assembly 8088	CMP x,y JXX διεύθυνση

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

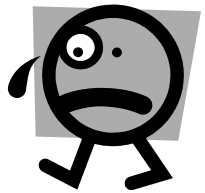
ΕΝΤΟΛΗ ΣΥΓΚΡΙΣΗΣ

■ Η εντολή CMP και οι τρόποι σύνταξής της

Σύγκριση 2 αριθμών, CMP a,b → a-b και επηρεάζει τα flags

- CMP καταχωρητής1, καταχωρητής2 — CMP AX,BX
- CMP καταχωρητής, [θέση μνήμης] — CMP AX,[200]
- CMP [θέση μνήμης], καταχωρητής — CMP [200],CX
- CMP καταχωρητής, τιμή — CMP DX, 1AF4
- CMP BY[διεύθυνση μνήμης], τιμή 8bit — CMP BY[200], 7A
- CMP WO[διεύθυνση μνήμης], τιμή 16bit — CMP WO[200], 10F5

~~CMP [θέση μνήμης], [θέση μνήμης]~~



ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΗ ΑΛΜΑΤΟΣ ΥΠΟ ΣΥΝΘΗΚΗ

Οι εντολές JXX και οι τρόποι σύνταξής τους (29 εντολές)

Άλμα υπό συνθήκη, JXX a → κάνε άλμα στην διεύθυνση a αν ισχύει η συνθήκη

- JE a (Jump on Equal)
  Κάνε άλμα στη διεύθυνση a αν το αποτέλεσμα της προηγούμενης πράξης 0
- JNE a (Jump on Not Equal)
  Κάνε άλμα στη διεύθυνση a αν το αποτέλεσμα της προηγούμενης πράξης διάφορο του 0
- JL a (Jump on Less)
  Κάνε άλμα στη διεύθυνση a αν στην προηγούμενη CMP ο $1^{ος}$ αριθμός < $2^{ου}$ αριθμού
- JLE a (Jump on Less or Equal)
  Κάνε άλμα στη διεύθυνση a αν στην προηγούμενη CMP ο $1^{ος}$ αριθμός ≤ $2^{ου}$ αριθμού
- JO a (Jump on Overflow)
  Κάνε άλμα στη διεύθυνση a αν overflow bit=1

κ.α. παρόμοιες εντολές (δες σημειώσεις)
Προσοχή: μέγιστο άλμα -128/+127 διευθύνσεις μνήμης από την JXX

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΗ ΑΛΜΑΤΟΣ ΑΝΕΥ ΣΥΝΘΗΚΗΣ

Η εντολή JMP και οι τρόποι σύνταξής της

Άλμα άνευ συνθήκης, JMP a

- JMP διεύθυνση μνήμης — JMP 0300
- JMP καταχωρητής — JMP BX
- JMP [διεύθυνση μνήμης] — JMP [0500]
- JMP segment:offset — JMP 0100:0400
- JMP FAR [διεύθυνση μνήμης] — JMP FAR[0600]

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 4

ΥΛΟΠΟΙΗΣΗ ΣΥΝΘΗΚΗΣ (IF)

Α/Α	Δομή της C	Κώδικας Assembly 8088	Παρατηρήσεις

1	Εντολή πριν ;
	if (συνθήκη)
	{
	Εντολή 1;
	Εντολή 2;
	...
	Εντολή ν;
	}
	Εντολή μετά ;
		Εντολή πριν
		CMP <παράμετροι>
		Jxx <διεύθυνση>
		Εντολή 1
		Εντολή 2
		...
		Εντολή ν
		<διεύθυνση> Εντολή μετά
			Η εντολή Jxx είναι αντίθετη από την (συνθήκη) της C. Π.χ. για την συνθήκη :
			if (ax= =0)
			η εντολή θα είναι :
			JNZ

2	Εντολή πριν ;
	if (συνθήκη)
	{ Εντολή 1;
	...
	Εντολή ν;}
	else
	{ Εντολή ν+1;
	...
	Εντολή ì ;}
	Εντολή μετά ;
		Εντολή πριν
		CMP <παράμετροι>
		Jxx <διεύθυνση>
		Εντολή ν+1(περίπτωση else)
		...
		Εντολή ì
		JMP <συνέχεια>
		<διεύθυνση> Εντολή 1 (περίπτωση if)
		...
		Εντολή ν
		<συνέχεια> Εντολή μετά
			Η εντολή Jxx είναι όμοια με την (συνθήκη) της C. Π.χ. για την συνθήκη :
			if (ax= =0)
			η εντολή θα είναι :
			JZ

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 4

ΑΣΚΗΣΗ 4.1

Γράψτε ένα πρόγραμμα που θα ταξινομεί δύο **μη προσημασμένους 16 bit** αριθμούς κατ' αύξουσα σειρά. Οι δύο αριθμοί περιέχονται στις θέσεις μνήμης **0200-0201** ο πρώτος και **0202-0203** ο δεύτερος. Εάν ο πρώτος αριθμός είναι μεγαλύτερος από τον δεύτερο τότε οι δύο αριθμοί θα εναλλάσσονται στη μνήμη. Εάν ο πρώτος είναι ίσος ή μικρότερος από τον δεύτερο θα παραμείνουν όπως είναι. Στο τέλος η θέση μνήμης **0204** να περιέχει μετά την εκτέλεση της σύγκρισης έναν από τους τρεις αριθμούς 0,1,2 όπως παρακάτω :

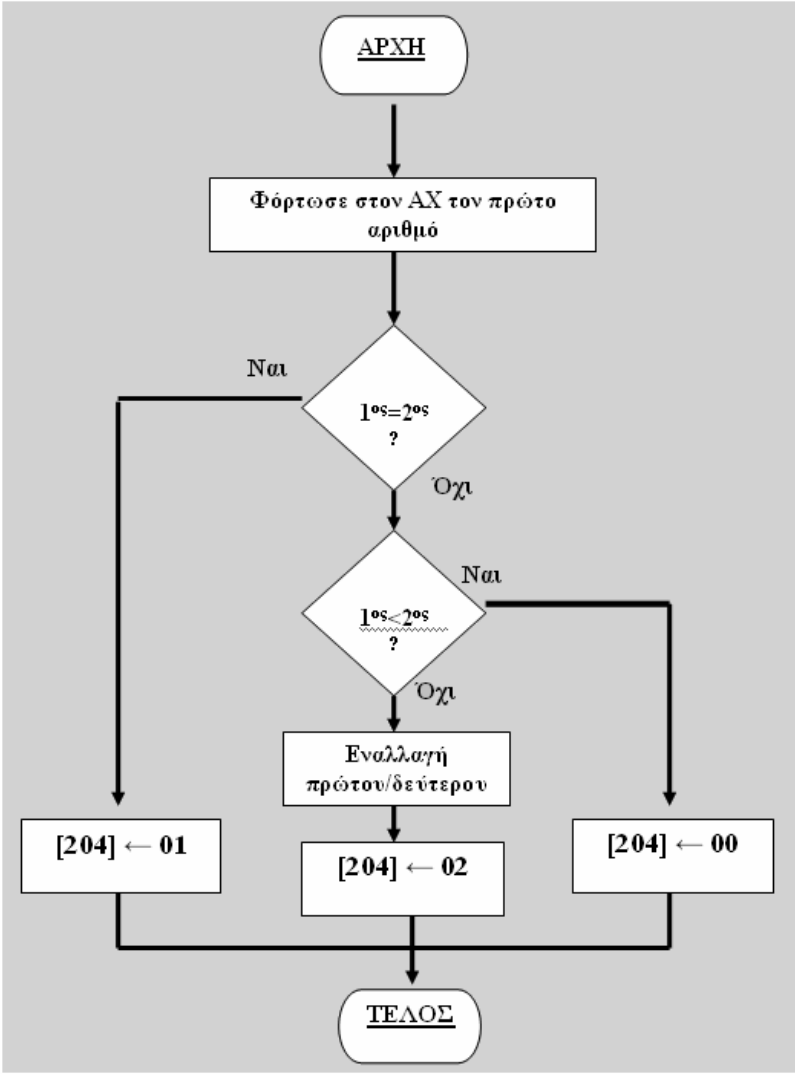
0 εάν ο πρώτος είναι μικρότερος από τον δεύτερο.

1 εάν οι δύο αριθμοί είναι ίσοι.

2 εάν ο πρώτος είναι μεγαλύτερος από τον δεύτερο.

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 4



ΑΣΚΗΣΗ 4.1

0200-0201	0202-0203	0204
23 45	12 34	02
AA AA	BB BB	00
FF FF	FF FF	01

4523 **>** 3412

AAAA **<** BBBB

FFFF **=** FFFF

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΑΠΛΟΣ ΒΡΟΧΟΣ n ΕΠΑΝΑΛΗΨΕΩΝ

```
                ……….
                MOV AL, 00          ; Αρχικοποίηση μετρητή επαναλήψεων
ετικέτα         ……….
                ……….                Εντολές προς επανάληψη
                ……….
                INC AL              ; AL=AL+1, αύξηση μετρητή επαναλήψεων
                CMP AL , n          ; Έλεγχος για τέλος επαναλήψεων
                JNE ετικέτα         ; Άλλη μια επανάληψη
                ……….
```

INC καταχωρητής → καταχωρητής=καταχωρητής+1

DEC καταχωρητής → καταχωρητής=καταχωρητής-1

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΑΣΚΗΣΗ 4.2

Γράψτε ένα πρόγραμμα που θα γράφει αυτόματα στη μνήμη και στις θέσεις από την 0200 και μετά τους αριθμούς 00, 01, 02, 03, ... με χρήση ενός απλού βρόχου επανάληψης. Κάθε αριθμός καταλαμβάνει 1 byte.

- Αρχικά εκτελέστε την επανάληψη 8 φορές
- Στη συνέχεια μετατρέψτε το πρόγραμμα ώστε να εκτελείται τόσες φορές όσο το περιεχόμενο του καταχωρητή CL
- Τέλος μετατρέψτε το πρόγραμμα ώστε να εκτελείται τόσες φορές όσο η τιμή της μεταβλητής που βρίσκεται στη θέση 0300 (1 byte)
- Πώς μπορούμε να μετατρέψουμε το πρόγραμμα ώστε να λειτουργεί για πίνακα μεταβλητής διεύθυνσης αρχής ?

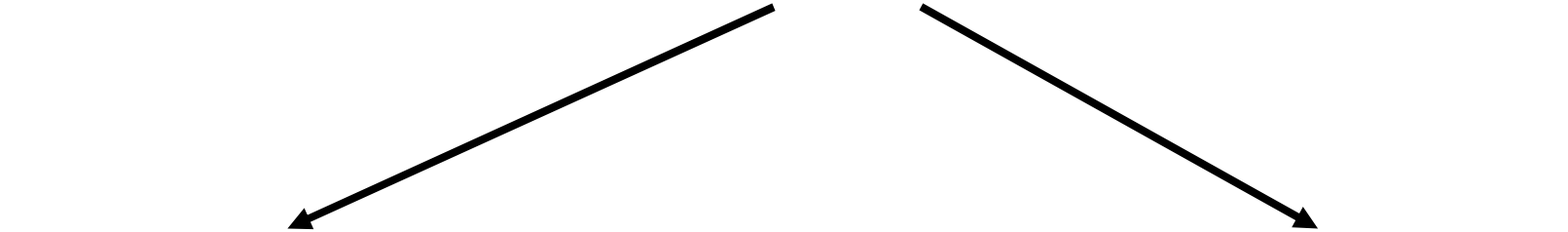
200	201	202	203	204	205	206	207	Θέση μνήμης
00	01	02	03	04	05	06	07	Περιεχόμενο

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 5

ΔΗΜΙΟΥΡΓΙΑ ΒΡΟΧΩΝ ΣΤΗΝ **ASSEMBLY**

ΒΡΟΧΟΙ



ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΗ LOOP

Η εντολή LOOP

LOOP διεύθυνση

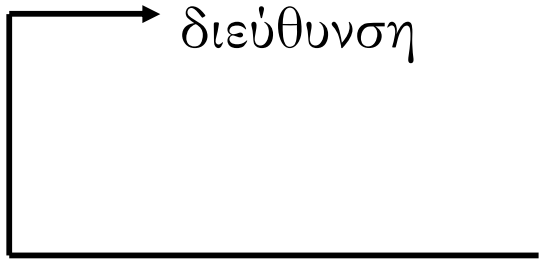
```
DEC CX
CMP CX,00
JNE διεύθυνση
```

Μειώνει τον CX κατά 1 και κάνει άλμα στη διεύθυνση αν ο CX ≠ 0

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΗ **LOOP**

Βρόχος n επαναλήψεων με LOOP



```
.....
MOV CX, n
διεύθυνση  ..... ; εντολές που
           ..... ; επαναλαμβάνονται
           LOOP διεύθυνση
.....
```

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΝΤΟΛΗ LOOP

Η εντολή LOOPZ

LOOPZ διεύθυνση

- Μειώνει τον CX κατά 1 και κάνει άλμα στη διεύθυνση αν ο CX ≠ 0 ΚΑΙ Zero flag=1

Η εντολή LOOPNZ

LOOPNZ διεύθυνση

- Μειώνει τον CX κατά 1 και κάνει άλμα στη διεύθυνση αν ο CX ≠ 0 ΚΑΙ Zero flag=0

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΔΕΙΚΤΟΔΟΤΟΥΜΕΝΗ ΠΡΟΣΠΕΛΑΣΗ

Διευθυνσιοδότηση [διεύθυνση]	π.χ. MOV [200] , CX
Διευθυνσιοδότηση [διεύθυνση + καταχ1]	π.χ. MOV [200+DI] , DX
Διευθυνσιοδότηση [διεύθυνση + καταχ1 + καταχ2]	π.χ. MOV [200+DI+BX], AL
Διευθυνσιοδότηση [καταχ]	π.χ. MOV [BX], AL
Διευθυνσιοδότηση [καταχ1 + καταχ2]	π.χ. MOV [BX+SI], AL

- Πλεονεκτήματα: Εύκολη και ευέλικτη σάρωση πινάκων στη μνήμη με βρόχους
- Καταχωρητές μόνον SI, DI, BX, BP
- 24 διαφορετικοί τρόποι σύνταξης

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 5

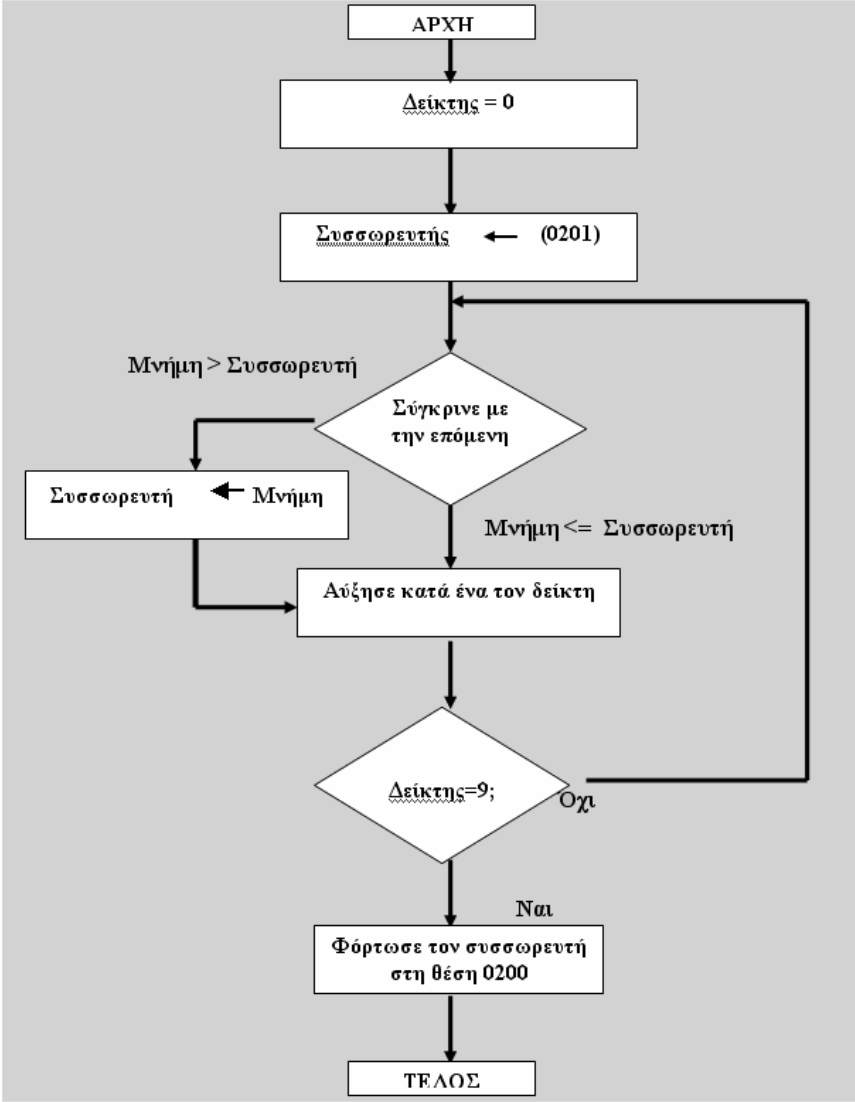
ΑΣΚΗΣΗ 5.1

Γράψτε το πρόγραμμα που βρίσκει τον μεγαλύτερο 8-bit μη προσημασμένο αριθμό από δέκα αριθμούς που βρίσκονται στις θέσεις μνήμης από **0201** έως και **020A** και βάλτε τον στη θέση μνήμης **0200**. Ο πίνακας των αριθμών δίνεται παρακάτω.

Θέση Μνήμης	Περιεχόμενα
0201	A2
0202	00
0203	40
0204	40
0205	0B
0206	5A
0207	FA
0208	55
0209	06
020A	4F

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 5



ΑΣΚΗΣΗ 5.1

Θέση Μνήμης	Περιεχόμενα
0200	FA
0201	A2
0202	00
0203	40
0204	40
0205	0B
0206	5A
0207	FA
0208	55
0209	06
020A	4F

ΤΕΙ ΣΕΡΡΩΝ
Τμήμα Πληροφορικής & Επικοινωνιών
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΑΣΚΗΣΗ 5.2

Να γραφεί πρόγραμμα που να προσθέτει, με χρήση βρόχου, τους αριθμούς 1+2+3+4+5 και να τοποθετεί το αποτέλεσμα στη διεύθυνση μνήμης **0200**

ΑΣΚΗΣΗ 5.3

Να τροποποιήσετε το ανωτέρω πρόγραμμα ώστε να προσθέτει τους αριθμούς 1+2+…+N όπου το N να είναι μεταβλητή και να βρίσκεται στη θέση μνήμης **0200**. Το αποτέλεσμα να τοποθετείται στην διεύθυνση μνήμης **0201-0202**