

Prénom Nom
Groupe

TP3 Assembleur : indirection

P. Carreno - P. Portejoie

Exercices 3-1 à 3-4

- Reprenez les exercices 3-1 à 3-4 du TD et faites-les fonctionner.

3-1 :

AL 00000010 02 +002	IP 00011100 1C +028
BL 00000010 02 +002	SP 10111111 BF -065
CL 01100100 64 +100	SR 00000000 00 +000
DL 00000000 00 +000	ISOZ

☐ Write Run Log ☐ Log Assembler

Source Code	List File	Configuratio
CLO		
MOV AL, [64]		
MOV BL, [65]		
CMP AL, BL		
JNS SUP		
INF:	MOV CL, 64	
JMP SUITE		
SUP:	MOV CL, 65	
SUITE:	MOV AL, [CL]	
INC AL		
MOV [CL], AL		
ORG 64		
DB 1		
DB 2		
END		

3-2 :

AL 00000000 00 +000	IP 00011000 18 +024		A				CLO	Assemble	S
BL 01101001 69 +105	SP 10111111 BF -065		A	H				Step	F
CL 00000000 00 +000	SR 00000010 02 +002		B		K	N		Run F9	S
DL 00000000 00 +000	ISOZ								

☐ Write Run Log ☐ Log Assembler Activity

Source Code | List File | Configuration | Tokens | Run Log

```

CLO
MOV BL, 64
BOUCLE: IN 00
CMP AL, 0D
JZ FIN
MOV[BL], AL
INC BL
JMP BOUCLE
FIN: MOV AL, 0
MOV [BL], AL
END

```

RAM ASCII View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	p	ð		d	a		û			Á		Ô				x
10	À	ô	ð			Ô										
20																
30																
40																
50																
60																
70																
80																
90																
A0																
B0																
C0																
D0																
E0																

3-3 :

AL 01010011 53 +083	IP 01000101 45 +069	C L O	Assemble	Slower
BL 01100011 63 +099	SP 10111111 BF -065	Step	Faster	
CL 11001011 CB -053	SR 00001000 08 +008	Run F9	STOP	
DL 00000000 00 +000	ISOZ			

☐ Write Run Log ☐ Log Assembler Activity

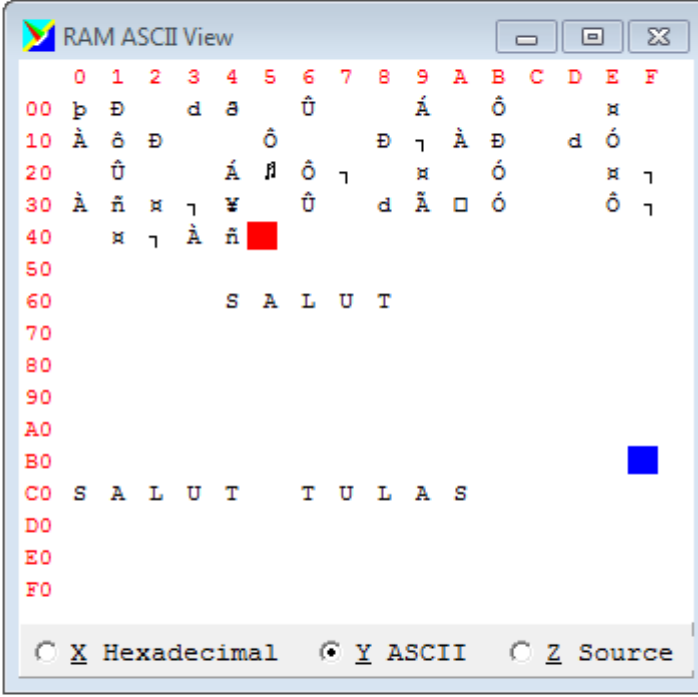
Source Code | List File | Configuration | Tokens | Run Log

```

CLO
MOV BL,64
BOUCLE: IN 00
CMP AL,0D
JZ FIN1
MOV[BL],AL
INC BL
JMP BOUCLE
FIN1:  MOV AL,0
      MOV [BL],AL

      MOV CL,C0
      MOV BL,64
      MOV AL,[BL]
ENDROIT:
      CMP AL,0
      JZ SUITE
      MOV [CL],AL
      INC BL
      MOV AL,[BL]
      INC CL
      JMP ENDROIT
SUITE:
      INC CL
MOITIE:
      DEC BL
      CMP BL,64
      JS FIN2
      MOV AL,[BL]
      MOV [CL],AL
      INC CL
      JMP MOITIE
FIN2:  END

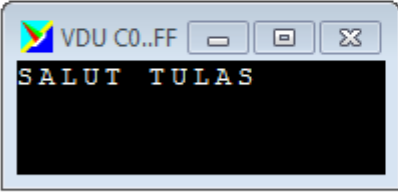
```



RAM ASCII View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	p	ð	d	ð	û				Á	Ô					x	
10	À	ô	ð		Ô			ð	¸	À	ð		d	Ó		
20	û			Á	ß	Ô	¸		x		Ó				x	¸
30	À	ñ	x	¸	¥	û		d	Ã	□	Ó			Ô	¸	
40		x	¸	À	ñ											
50																
60					S	A	L	U	T							
70																
80																
90																
A0																
B0																
C0	S	A	L	U	T			T	U	L	A	S				
D0																
E0																
F0																

☐ Hexadecimal ☒ ASCII ☐ Source



VDU C0..FF

SALUT TULAS

3-4 :

The screenshot displays an 8086 assembler interface with the following components:

- Registers and Status:**
 - AL 00000000 00 +000
 - BL 11000101 C5 -059
 - CL 00000000 00 +000
 - DL 00000000 00 +000
 - IP 00100010 22 +034
 - SP 10111111 BF -065
 - SR 00000010 02 +002
 - ISOZ
- Control Panel:**
 - Buttons: Assemble, Slower, Copy, Step, Faster, Cpu, Run F9, STOP, Show.
 - Checkboxes: ☐ Write Run Log, ☐ Log Assembler Activity.
 - Tabs: Source Code, List File, Configuration, Tokens, Run Log.
- Source Code:**

```

CLO
MOV AL,0
PUSH AL
LIRE: IN 00
CMP AL,0D
JZ SUITE
PUSH AL
JMP LIRE
SUITE: MOV BL,C0
AFFICHE:
POP AL
CMP AL,0
JZ END
MOV [BL],AL
INC BL
JMP AFFICHE
END: END

```
- RAM ASCII View:**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	p	ð		à		a		û			Á	-	à		À	
10	÷	ð		À	á		û			Á	Ô				æ	
20	À	ô														
30																
40																
50																
60																
70																
80																
90																
A0																
B0																
C0	T	U	L	A	S											
D0																
E0																
F0																
- VDU CO..FF:**

```

TULAS

```

Exercice 3-5

- Modifiez le programme de saisie d'une chaîne de caractères de l'exercice 3-2 pour qu'il effectue la saisie à l'adresse décimale 192 et détermine si la chaîne saisie est un palindrome ou pas. La réponse affichée sera *OUI* ou *NON à la suite de la chaîne*.

Exemples : LAVAL et ELLE sont des palindromes.

Le résultat pour LAVAL sera : LAVAL OUI

Quelques petits conseils :

- vous pouvez fortement vous inspirer de ce que vous avez fait précédemment pour la saisie et l'affichage (code quasi ré-utilisable au prix de légères modifications) :
 - > saisie (+ mémorisation de la dernière adresse)
 - > palindrome ?
 - > Affichage selon résultat
- pour LAVAL la condition d'arrêt sur l'égalité des pointeurs est suffisante, pas pour ELLE ! Réfléchissez bien pour trouver la solution adaptée. Pensez à tester votre programme dans les 2 cas.
- enfin pour l'affichage vous considèrerez que le principe est unique et que seule change l'adresse d'implantation de la chaîne. Ainsi en sélectionnant le pointeur adéquat la chaîne correspondante sera affichée. Pour ce faire vous utiliserez en fin de code de votre programme :

```
ORG    A0
DB     "NON"
DB     0
DB     "OUI"
DB     0
END
```

Code :

```

CLO
MOV CL,C0

LIRE:  IN 00
      CMP AL,0D
      JZ D_ADRESSE
      MOV [CL],AL
      INC CL
      JMP LIRE

D_ADRESSE:
      PUSH CL
      MOV DL,C0
      DEC CL

PALINDROME:
      CMP CL,DL
      JNS OUI
      MOV AL,[CL]
      MOV BL,[DL]
      CMP AL,BL
      JNZ NON
      DEC CL
      INC DL
      JMP PALINDROME

OUI:
      MOV AL,A4
      POP CL
      INC CL
      JMP AFFICHAGE

NON:
      POP CL
      INC CL
      MOV AL,A0

AFFICHAGE:
      MOV BL,[AL]
      CMP BL,0
      JZ END
      MOV [CL],BL
      INC CL
      INC AL
      JMP AFFICHAGE

END:
      ORG A0
      DB "NON"
      DB 0
      DB "OUI"
      DB 0
      END

```

