TD 4

Exercice 1 : Exemple d'application en mode 0

On donne le schéma de la figure 1,

A) on veut écrire un programme qui permet de faire clignoter les diodes Led Jusqu'à l'appui sur SW0

B) On veut écrire un programme qui affiche les chiffres de 0 à 15 sur les 7 segments.

C) On veut écrire un programme qui permet de faire clignoter les diodes paires si on appuie sur SW0 et les diodes impaires si on appuie sur SW1

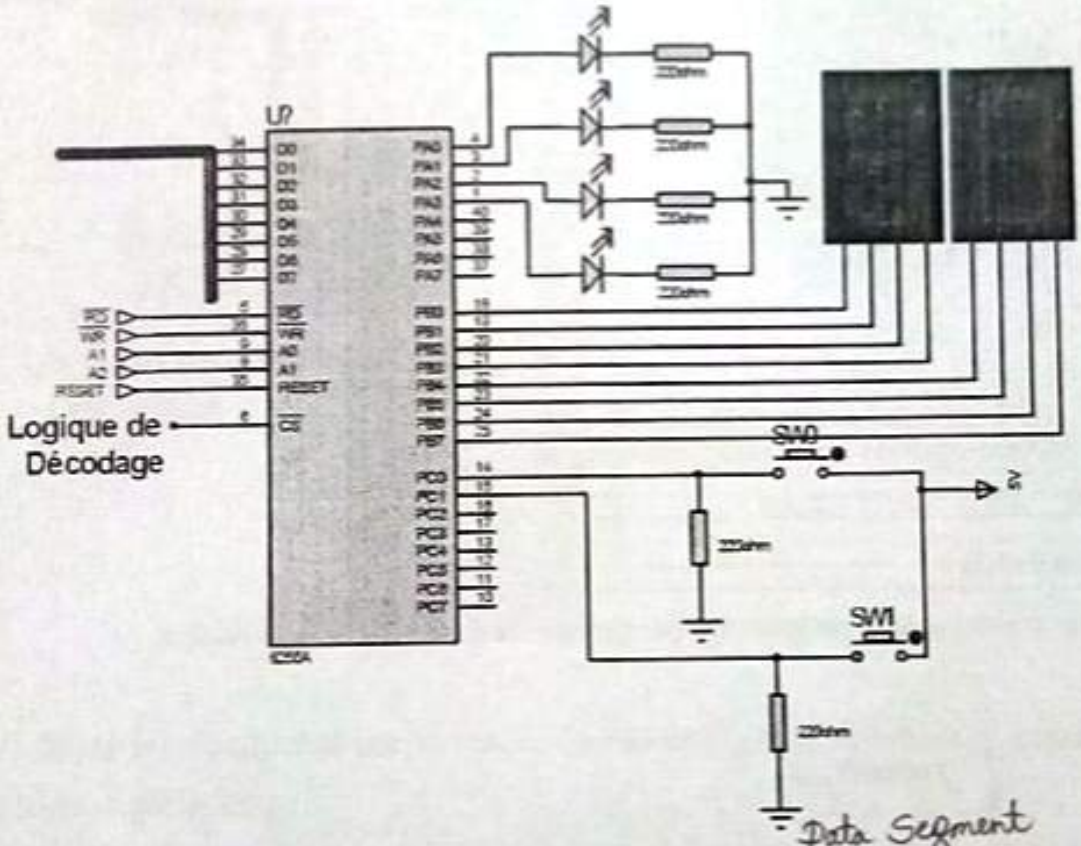


Figure 1

On suppose que les adresses des ports est comme suit :

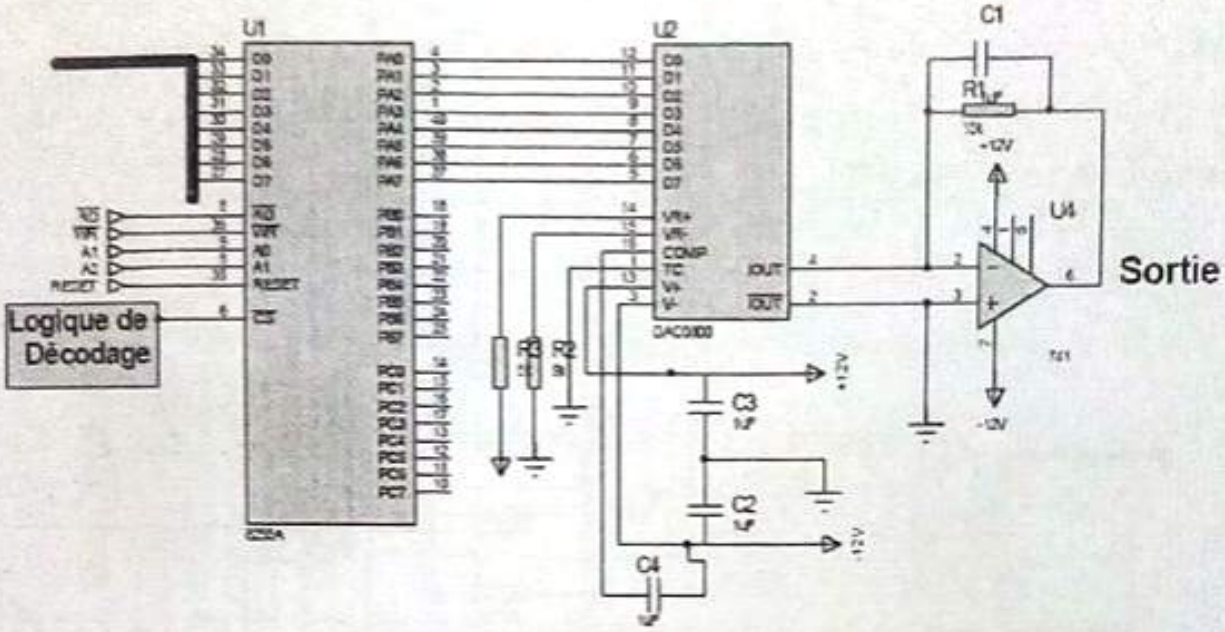
Port A : 300H Port B : 302H Port C : 304H

Registre de commande : 306H

```
Data Segment
    DB
    DB
Data ENDS
Code Segment
    assume cs:code
main: mov ax, 4C00H
    int 21h
code ENDS
END Main
```

Exercice 2 :

On donne le schéma de la figure suivante :

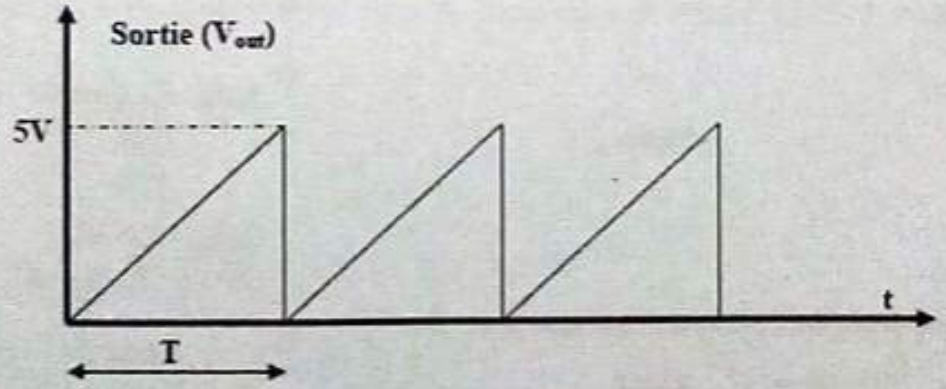


On suppose que les adresses des ports est donner comme suit :

Port A : 300H Port B : 302H Port C : 304H

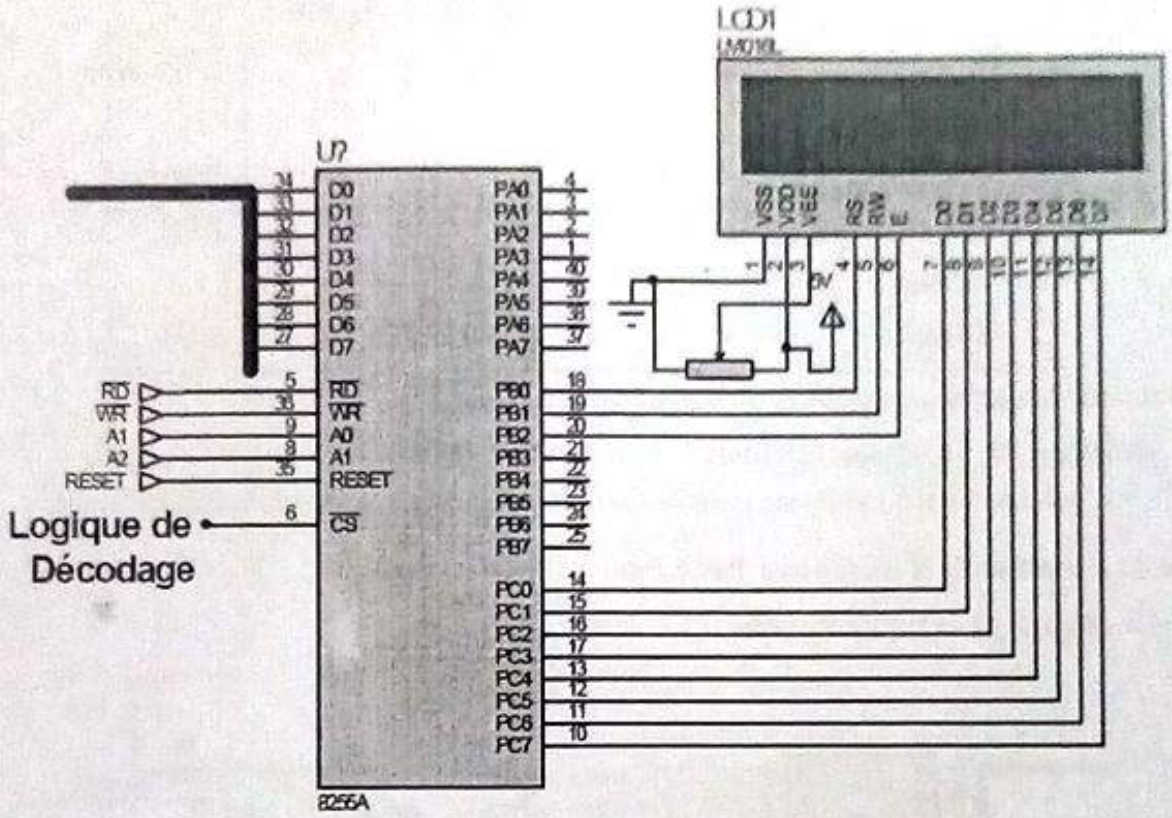
Registre de commande : 306

Donner l'organigramme ainsi que le programme qui permet de générer le signal suivant :



Exercice 3 :

Pour commander un afficheur LCD 16*2 (16 caractères , 2 lignes) on propose le schéma du montage suivant :



Un LCD est formé essentiellement par un bus de donnée de 8 bits et un bus de commande et contrôle formé par trois pines(E, Rd/Rw, RS) :

E : entrée de validation un front descendant sur cette pine provoque la validation de la donnée ou de la commande.

RS : elle permet de distinguer les commandes et les données. RS = 0 le bus D0-D7 accepte des commandes

Commande	Code hexa
Effacement LCD	01H
Home	0EH
Direction vers la droite	06H

RS = 1 le bus D0-D7 accepte des donnéesEnfin Rd/Rw : c'est pour donner l'ordre de lecture ou écriture sur LCD. Parmis les commandes on trouve :

Exemple de programme qui affiche le message 'bonjour iset n' sur LCD

Avant de commencer le programme il faut déterminer les mots qu'il faut envoyer au portB pour valider une donnée ou valider une commande d'où le tableau suivant :

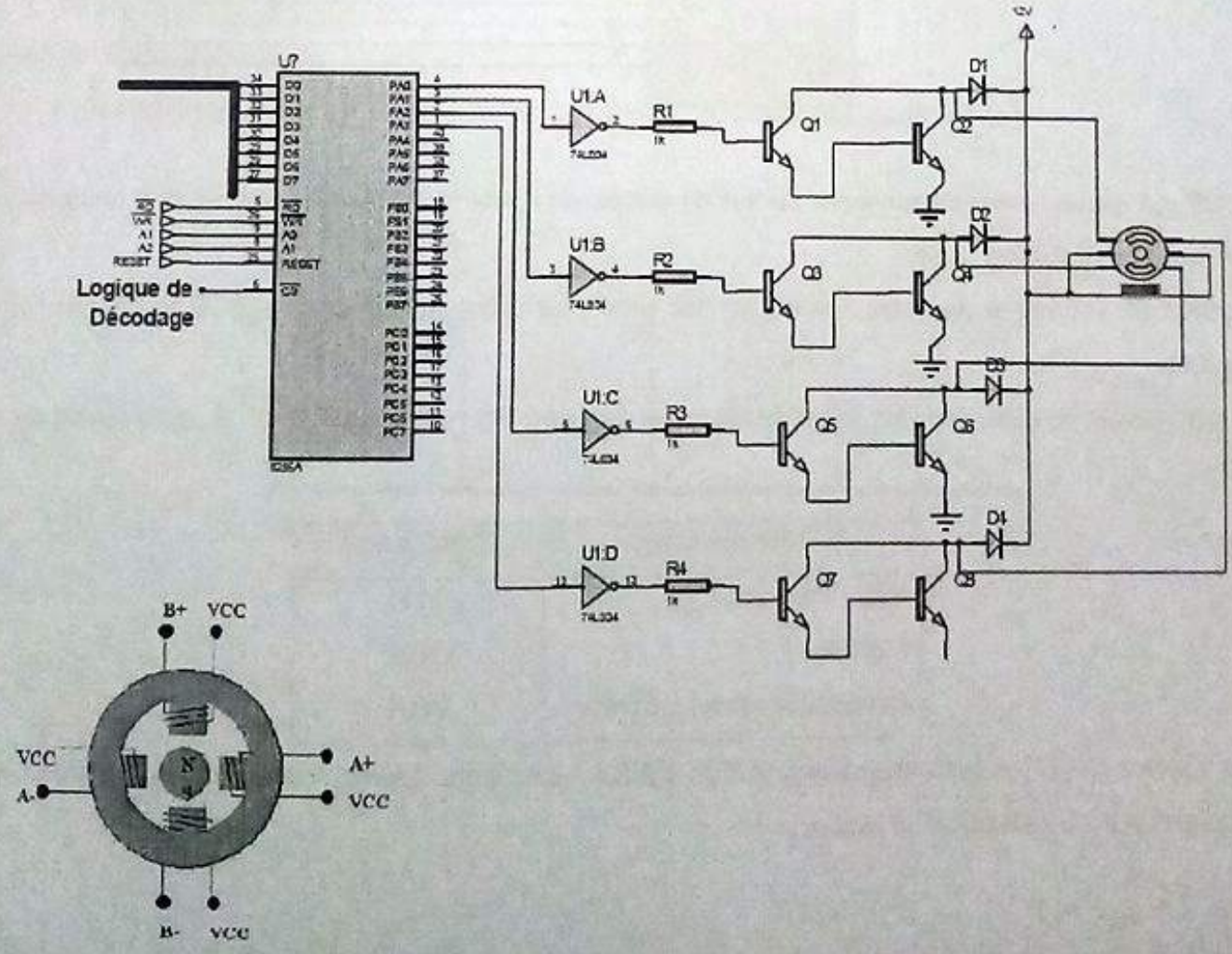
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	HEXA
	X	X	X	X	X	E	R/W	RS	
Validation d'une commande	0	0	0	0	0	1	0	0	04H
	0	0	0	0	0	0	0	0	00H
Validation d'une donnée	0	0	0	0	0	1	0	1	05H
	0	0	0	0	0	0	0	1	01H

On suppose que les adresses des ports est donner comme suit :

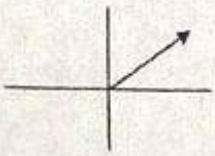
Port A : 300H Port B : 302H Port C : 304H

Exercice 4 : Commande d'un moteur Pas à Pas

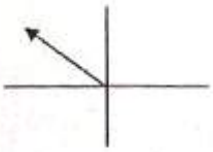
On donne le schéma de la figure suivante :



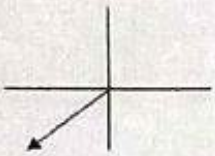
Les différentes phases sont les suivantes (en mode avancement par un pas):



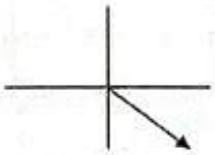
Phase 1



Phase 2

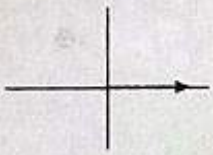


Phase 3

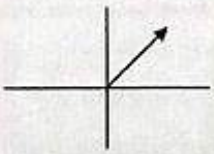


Phase 4

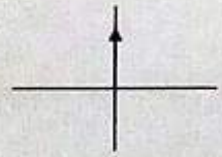
Les différentes phases sont les suivantes (en mode avancement par demi pas):



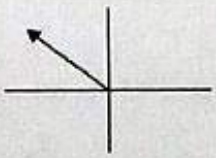
Phase 1



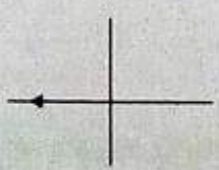
Phase 2



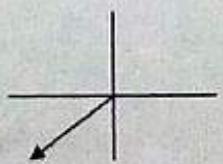
Phase 3



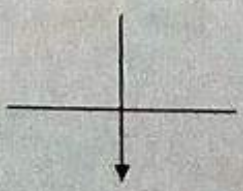
Phase 4



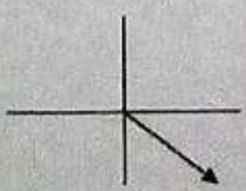
Phase 5



Phase 6



Phase 7



Phase 8

La commande des phases du moteur pas à pas bipolaire a aimant permanent est donnée par le tableau suivant :

➢ pas complet

Phase	1	2	3	4	5	6	7	8
Phase 1 (A+)	1	0	0	1	1	0	0	1
Phase 2 (B+)	1	1	0	0	1	1	0	0
Phase 3 (A-)	0	1	1	0	0	1	1	0
Phase 4 (B-)	0	0	1	1	0	0	1	1

➢ Demi pas

Phase	1	2	3	4	5	6	7	8
Phase 1 (A+)	1	0	0	0	1	0	0	0
Phase 2 (B+)	0	1	0	0	0	1	0	0
Phase 3 (A-)	0	0	1	0	0	0	1	0
Phase 4 (B-)	0	0	0	1	0	0	0	1

D'où le programme par exemple qui fait tourner le moteur pas à pas de 90°

: On suppose que les adresses des ports est donner comme suit :

Port A : 300H Port B : 302H Port C : 304H

Registre de commande : 306H 1 pas = 0.9°

A+ est connecté avec PA0 B+ est connecté avec PA1 A- est connecté avec PA2 B- est connecté avec PA3

TD2

Exercice 1

```
mov AX, 03H
mov BX, 01H
ADD AX, BX
mov BX, 02H
ADD AX, BX
```

Exercice 2

```
mov AX, 04H
push AX
mov AX, 01H
POP AX
```

Exercice 3

```
mov AX, 03H
mov BX, 01H
SUB AX, BX
```

Exercice 4

```
mov AX, 01H
mov BX, 02H
MUL BX
```

Exercice 5

```
mov AX, 06H
mov BX, 03H
Push AX
Push BX
POP AX
POP BX
```

Exercice 6

```
mov AX, 02H
MUL AX
```

TD3

Exercice 1

```
mov AX, 06H
mov BX, 09H
CMP AX, BX
jl cas1
jg cas2
cas1:
    mov DX, 02H
jmp fin
cas2:
    mov DX, 01H
```

Exercice 2

```
mov AX, 06H
Test AX, 1
jz pair
jmp impair
pair:
    mov DX, 0 H
    jmp end
impair:
    mov DX, 01H
end:
```

Exercice 3

```
mov AH, 04H
mov AL, 06H
mov BH, 07H
mov BL, 08H
cmp AH, AL
cmp BH, BL
jg cas1
jL cas2
cas1:
    mov DL, 01H
cas2:
    mov DL, 02H
```

ou bien :

```
cmp AH, AL
jg cas1
cmp BH, BL
jg cas1
cas1:
    mov DL, 01H
end
```

Exercice 4

```
mov AX, 04H
mov BX, 07H
mov CX, 08H
exercice:
Push AX
Push BX
loop exercice
```

Exercice 5

```
mov AX, 01H
mov BX, 02H
exercice:
    ADD BX, 06H
    inc AX
    cmp AX, 0AH
    jl end
    jmp exercice
end:
```

Exercice 6

```
mov CX, 05H
mov AX, 01H
etq:
    MUL CX
    loop etq
```

Exercice 7:

```
mov AX,
mov BX,
Cmp AX, BX
je etq1
mov BX, AX
jmp end
etq1:
    ADD BX,05H
end:
```

Exercice 8:

```
mov AH, 02H
mov DX, 'P'
int 21H
```

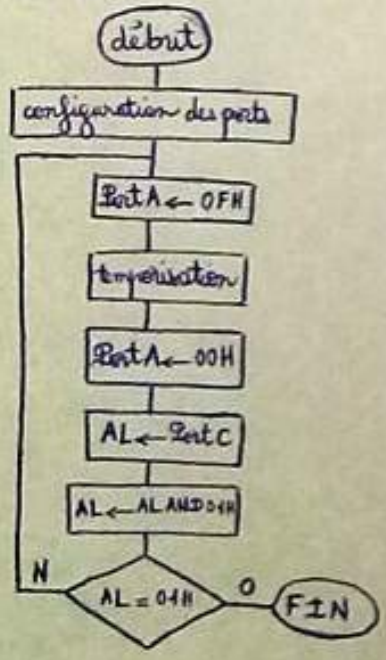
Exercice 9:

```
mov CX, 04H
mov BX, 01H
ADD CX, BX
mov AH, 02H
ADD CX, 03H
mov DX, CX
int 21H
```

TD4:

Exercice 1: DIODES. LED

- organigramme:

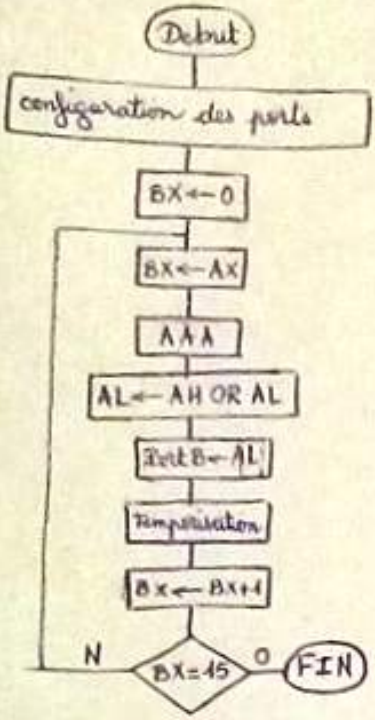


- programme:

```
Donnee Segment
PortA EQU 300H
PortC EQU 302H
Reg_com EQU 306H
Mot_com EQU 91H
Masque_SBO EQU 01H
Diode_allume EQU 0FH
Diode_eteinte EQU 00H
Donnee ENDS
Code Segment
Assume CS: code, DS: donnee
Prog Proc
        mov AX, donnee
        mov DX, AX
        mov AL, Mot_com
        out Reg_com, AL
Debut:  mov AL, Diode_allume
        out PortA, AL
        call temp 0
        mov AL, Diode_eteinte
        out PortA, AL
        call Temp 0
        IN AL, PortC
        AND AL, 01H
        cmp AL, 01H
        jnz Debut
        mov AX, 4C00H
        int 21H
        Prog ENDP
Temp 0: mov CX, FFFFH
Temp 1: Push CX
        mov CX, FFFFH
Temp 2: NOP
        NOP
        NOP
        NOP
        Loop Temp 2
        POP CX
        Loop Temp 1
        RET Code
        ENDS
END Prog
```

Afficheur 7 segments

*organigramme :

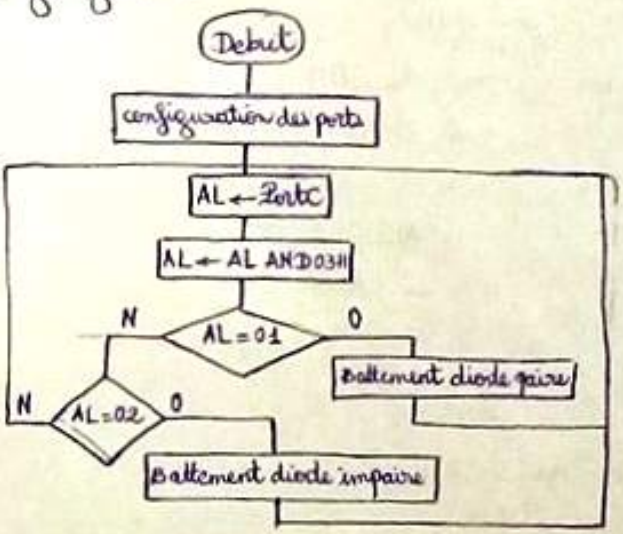


*programme :

```
Donnee Segment
PortA EQU 300H
PortB EQU 302H
Reg_com EQU 306H
Mot_com EQU 91H
Masque_5100 EQU 01H
Diode_allume EQU 0FH
Diode_eteinte EQU 00H
Donnee ENDS
Code Segment
Assume CS: code, DS: donnee
Prog Proc
        mov AX, donnee
        mov DS, AX
        mov AL, Mot_com
        out Reg_com, AL
        XOR BX,BX
Debut: mov AX,BX
        AAA
        OR AL,AH
        out PortB,AL
        call Temp0
        inc BX
        cmp BX,15
        jnz Debut
        mov AX,4C00H
        int 21H
        Prog ENDP
Temp0 : mov CX, 7FFFH
Temp1 : Push CX
        mov CX,7FFFH
Temp2 : NOP
        NOP
        NOP
        NOP
        Loop Temp2
        POP CX
        Loop Temp1
        RET
        Code ENDS
END Prog
```

Diodes Paires - Diodes Impaires

*organigramme :



*Programme :

```
Donnee Segment
PortA EQU 300H
PortC EQU 304H
Rg_com EQU 306H
Mot_com EQU 81H
Masque_5100 EQU 01H
Diode_allume EQU 0FH
Diode_eteinte EQU 00H
Donnee ENDS
Code Segment
Assume CS: code, DS: donnee
Prog Proc
```

```
        mov AX, donnee
        mov DS, AX
        mov AL, Mot_com
        out Reg_com AL
Debut:  in AL, PortC
        AND AL, 03H
        cmp AL, 01H
        jz Diode_paire
        cmp AL, 02H
        jz Diode_impaire
        jmp Debut

Diode_paire: mov AL, 05H
             out PortA, AL
             call Temp 0
             mov AL, 00H
             out PortA, AL
             call Temp 0
             jmp Debut
Diode_impaire: mov AL, 0AH
               out PortA, AL
               call Temp0
               mov AL, 00H
               out PortA, AL
               call Temp0
               jmp Debut

    mov AX, 4C00H
    int 21H
    Prog ENDP

Temp0: mov CX, FFFFH
Temp1, Push CX
       mov CX, FFFFH
Temp2: NOP
       NOP
       NOP
       NOP
       Loop Temp2
       POP CX
       Loop Temp 1
       RET
       Code ENDS
END Prog
```


* Exercice 3 : * Afficheur LCD *

```
Donnee Segment
Message db 'bonjour ixet n'
PortB EQU 302H
PortC EQU 304H
Reg_com EQU 306H
Mot_com EQU 80H
Donnee ENDS
Code Segment
Assume CS : code, DS : donnee

Prog Proc
    mov AX, donnee
    mov DS, AX
    mov AL, Mot_com
    out Reg_com, AL
    LEA SI, message
    mov CX, 14H
    call Init_LCD

debut : mov AL, [SI]
        out PortC, AL
        call Vali-donnee
        inc SI
        loop debut
        init_LCD: mov AL, 01H
                  out PortC, AL
                  call vali_commande
                  mov AL, 0EH
                  out PortC, AL
                  call Vali_commande
                  mov AL, 06H
                  out PortC, AL
                  call Vali_commande
                  RET
    Vali_commande: mov AL, 04H
                   out PortB, AL
                   mov AL, 00H
                   out PortB, AL
                   RET
    Vali_donnee : mov AL, 05H
                  out PortB, AL
                  mov AL, 01H
                  out PortB, AL

    mov AX, 4C00H
    int 21H
    Prog ENDP
```

```
Code ENDS
END Prog
```

*** Exercice 4 :** *** Moteur Pas à Pas ***

```
Donnee Segment
Phase db 03H,06H,0CH,09H,03H,06H, 0CH, 09H
PortA EQU 300H
Reg_Com EQU 306H
Mot_com EQU 80H
Donnee ENDS
Code Segment
Assume CS: code, DS: donnee
Prog Proc
    mov DS,AX
    mov AL, Mot_com
    out Reg_com, AL
    mov CX, 100H
debut 1: LEA SI, phase
debut 2: mov AL, [SI]
         out PortA, AL
         inc SI
         cmp SI,7
         jnz debut 2
         loop debut 1
         mov AX, 4C00H
Prog ENDP
Code ENDS
END Proc
```