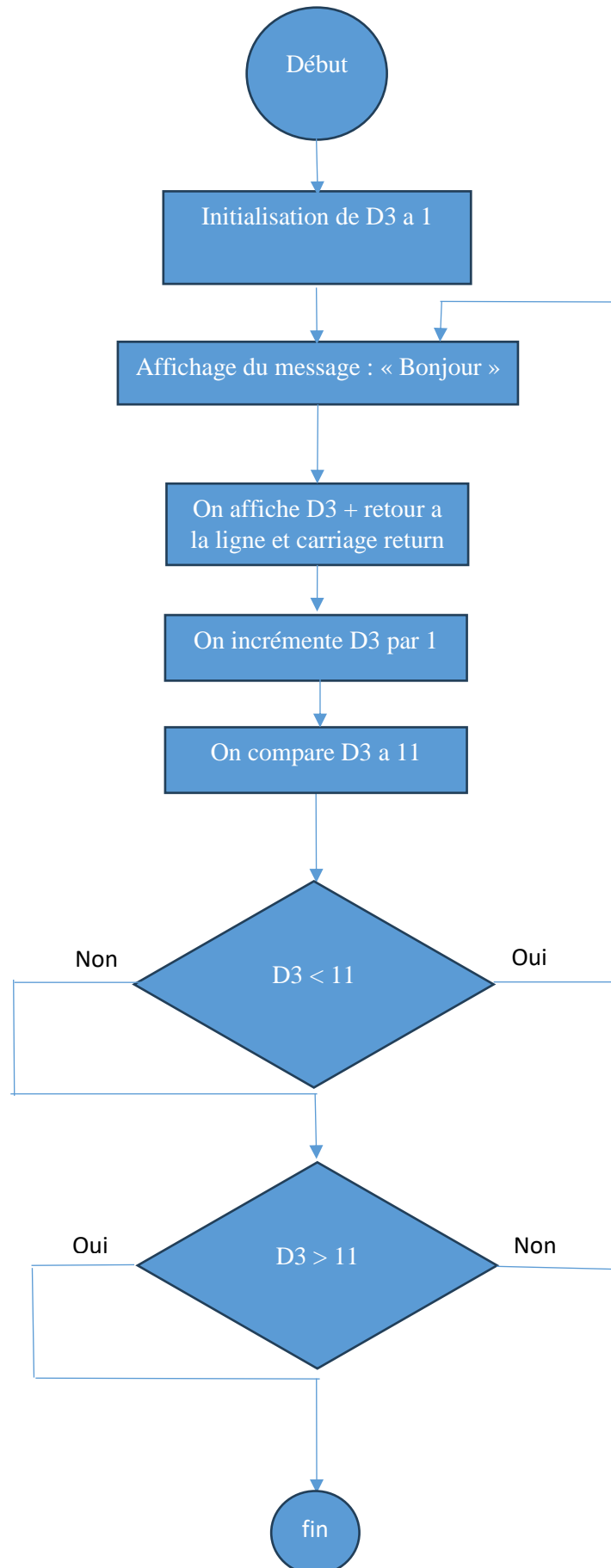


Compte rendu TP 2

Exercice 1 :

Organigramme :



Programme :

```
*-----*
* Title      : TP 2 EXO 1
* Written by : ERRARD - DACCACHE
* Date       : 13/03/24
* Description: TP 2
*-----*

    ORG      $1000
START:                                ; first instruction of program
CR EQU $0D                             *retour chariot (carriage return)
LF EQU $0A                             *Saut de ligne (Line Feed)
* Put program code here
    LEA MESSAGE, A1                    *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                      *Mettre 14 dans le registre D0
    TRAP #15                          *Affichage de MESSAGE
    MOVE.B #1, D3                      *On met 1 dans le registre de donne D3
*-----*
BOUCLE    LEA MESSAGE2, A1             *chargement de MESSAGE2 dans le registre
d'adress A1
    MOVE.B #14,D0                      *Mettre 14 dans le registre D0
    TRAP #15                          *Affichage de MESSAGE2
    MOVE.B #3,D0                      *on met 3 dans D0
    MOVE.B D3,D1                      *on met le contenu de D3 dans D1
    TRAP #15                          *on afficher le nombre contenu dans D1
    ADD #1,D3                         *on ajoute 1 a la valeur contenu dans D3
    CMP.L #11,D3                      *on compare 11 a D3
    BGT ENDER                        *si D3 est plus grand que 11 le code passe au
code contenu dans l'etiquette ENDER
    BLT BOUCLE                       *si D3 est plus petit que 11 on recommence le
code de l'etiquette BOUCLE

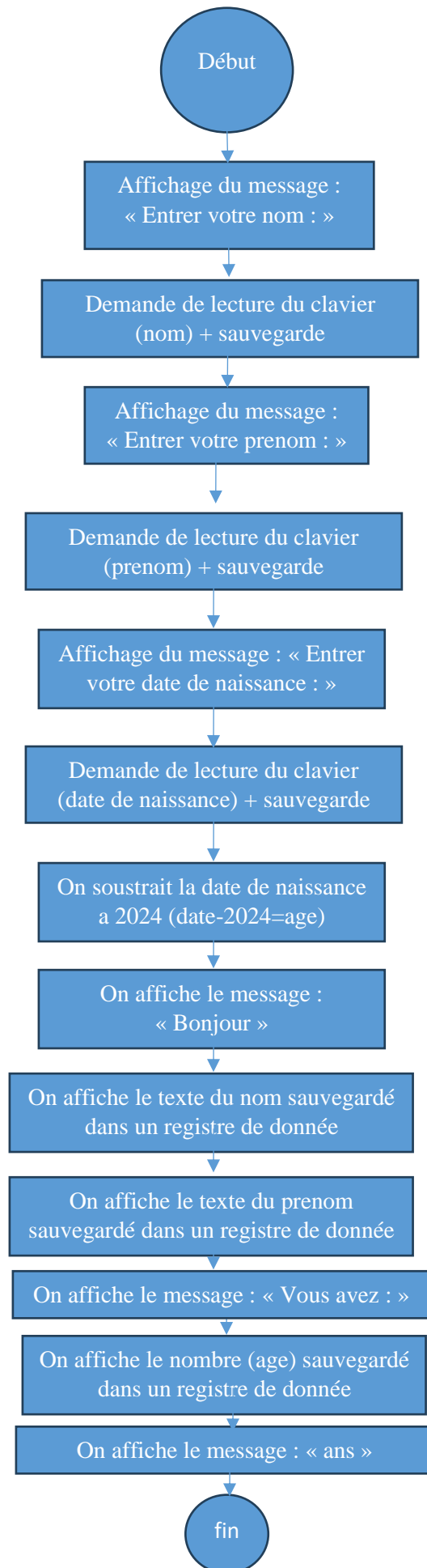
ENDER     MOVE.B #9,D0                 *on met 9 dans D)
    TRAP #15                         *on arrete le programme
    SIMHALT                          ; halt simulator

* Put variables and constants here
MESSAGE   DC.B '----- TP2 EXO 1 -----',CR,LF,0
MESSAGE2  DC.B CR,LF,'Bonjour ',0

    END      START                    ; last line of source
```

Exercice 2 :

Organigramme :



Programme :

```
*-----
* Title       :   TP2 EXO 2
* Written by  :   ERRARD - DACCACHE
* Date       :   13/03/24
* Description: EXO 2
*-----
        ORG      $1000
START:                                ; first instruction of program
CR EQU $0D                            *retour chariot (carriage return)
LF EQU $0A                            *Saut de ligne (Line Feed)
* Put program code here
        LEA MESSAGE0, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *Affiche le message
*-----
        LEA MESSAGE1, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *affichage de MESSAGE1
        MOVE.B #2,D0                  *on met 2 dans D0
        TRAP #15                      *on demande a l'utilisateur d'entrer une donner
sur le clavier
        MOVE.L A1,D2                  *on met la donne sauvegarder dans A1 dans le
registre de donne D2
        LEA MESSAGE2, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on affiche MESSAGE2
        MOVE.B #2,D0                  *on met 2 dans D0
        TRAP #15                      *on demande a l'utilisateur d'entrer une donner
sur le clavier
        MOVE.L A1,D3                  *on met la donne sauvegarder dans A1 dans le
registre de donne D3
        LEA MESSAGE3, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on affiche MESSAGE3
        MOVE.B #4,D0                  *on met 4 dans D0
        TRAP #15                      *on demande a l'utilisateur d'entrer un nombre
sur le clavier
        MOVE.L D1,D4                  *on met la donne sauvegarder dans D1 dans le
registre de donne D4
        LEA MESSAGE4, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on affiche MESSAGE4
        MOVE.L D2,A1                  *on met la donné qui est dans D2 dans A1
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on afficher ce qui dans A1 (le Nom)
        LEA MESSAGE7, A1              *chargement de message dans l'adresse de
registre A1                           ;
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on affiche MESSAGE7
        MOVE.L D3,A1                  *on met la donné qui est dans D3 dans A1
        MOVE.B #14,D0                 *Mettre 14 dans le registre D0
        TRAP #15                      *on afficher le contenu de A1 (le Prenom)
        LEA MESSAGE5, A1              *chargement de message dans l'adresse de
registre A1                           ;
```

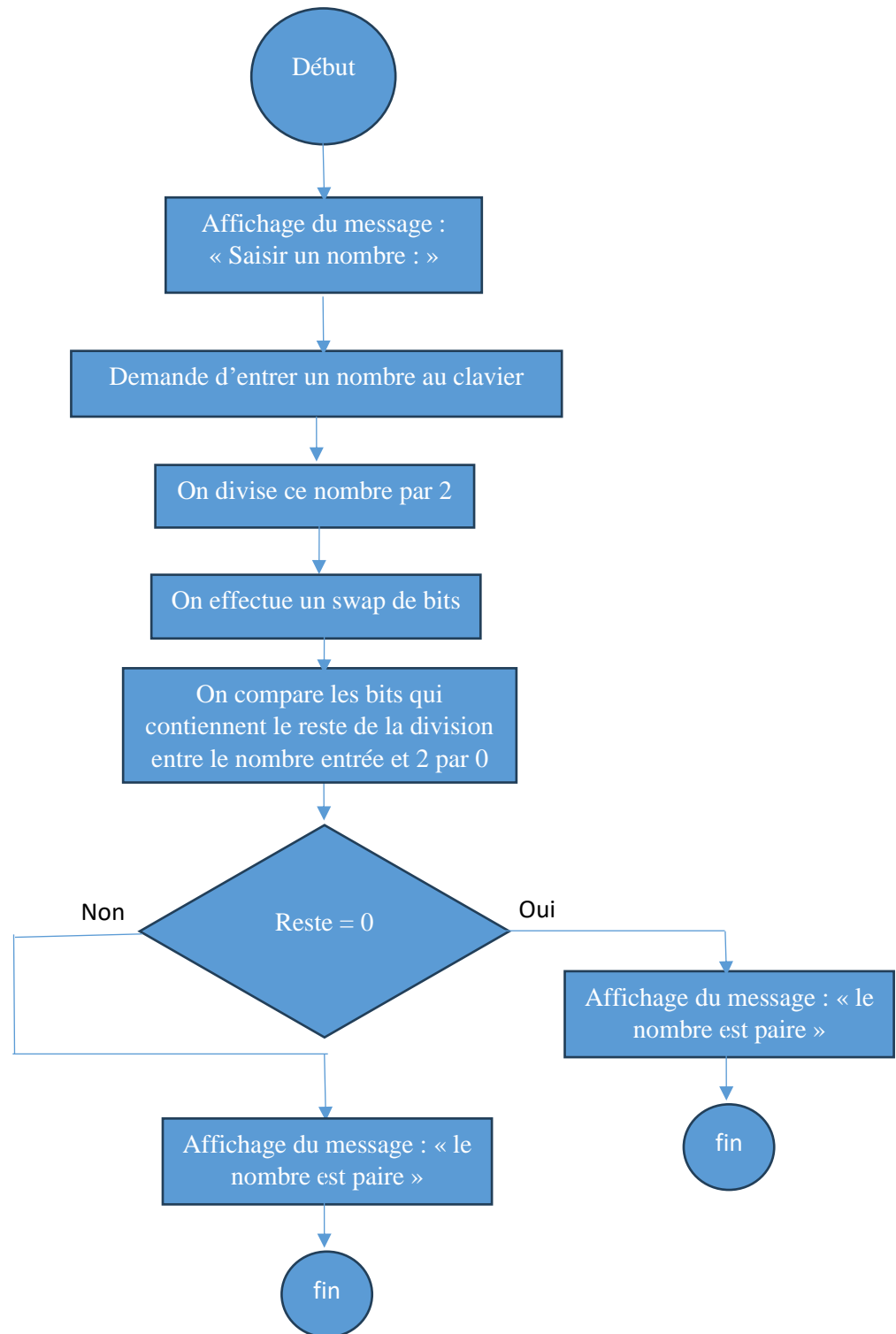
```
        MOVE.B #14,D0      *Mettre 14 dans le registre D0
        TRAP #15           *on affiche MESSAGE5
        MOVE.L #2024,D5    *on met 2024 dans D5
        SUB.L D4,D5        *on soustrait D5 (2024) par D4 (la date de
naissance donner par l'utilisateur)
        MOVE.L D5,D1       *on met le contenu de D5 dans D1
        MOVE.B #3,D0       *Mettre 14 dans le registre D0
        TRAP #15           *on afficher le contenu de D1 (le resultat de
la soustraction qui est l'age)
        LEA MESSAGE6, A1   *chargement de message dans l'adresse de
registre A1
        MOVE.B #14,D0      *Mettre 14 dans le registre D0
        TRAP #15           *on affiche MESSAGE6
        SIMHALT            ; halt simulator

* Put variables and constants here
MESSAGE0 DC.B '----- TP 2 EXO 2 -----',0,CR,LF
MESSAGE1 DC.B CR,LF,'Entrer votre nom: ',0
MESSAGE2 DC.B CR,LF,'Entrer votre prenom: ',0
MESSAGE3 DC.B CR,LF,'Entrer votre annee de naissance: ',0
MESSAGE4 DC.B CR,LF,'Bonjour ',0
MESSAGE5 DC.B '. Vous avez: ',0
MESSAGE6 DC.B ' ans!',0
MESSAGE7 DC.B ' ',0

        END      START      ; last line of source
```

Exercice 3 :

Organigramme :



Programme :

```
*-----*
* Title      :   EXO 3 TP 2
* Written by :   ERRARD - DACCACHE
* Date       :   13/03/24
* Description:   EXO 3
*-----*

    ORG      $1000
START:                                ; first instruction of program
CR EQU $0D                            *retour chariot (carriage return)
LF EQU $0A                            *Saut de ligne (Line Feed)
* Put program code here
    LEA MESSAGE0, A1                  *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                    *Mettre 14 dans le registre D0
    TRAP #15                         *Affiche le message
*-----*
    LEA MESSAGE1, A1                  *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                    *Mettre 14 dans le registre D0
    TRAP #15                         *Affiche MESSAGE1
    MOVE.B #4,D0                     *Mettre 4 dans le registre D0
    TRAP #15                         *demande a l'utilisateur d'insere un nombre
    DIVS #2,D1                       *on divise le nombre contenu dans D1 par 2
    SWAP D1                          *on echange les bites 1 a 4 avec les bites de 5
a 8
    CMP #0,D1                        *on compare 0 avec les bits du reste de la
division effectuee
    BEQ PAIRE                        *si le reste est egale a 0 on passe au code de
l'etiquette paire (donc le nombre est paire)
    BNE IMPAIRE                      *si le reste n'est pas egale a 0 on passe au
code de l'etiquette impaire (donc le nombre est impaire)
PAIRE LEA MESSAGE2, A1                *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                    *Mettre 14 dans le registre D0
    TRAP #15                         *Affiche MESSAGE2
    MOVE.B #9,D0                     *Mettre 14 dans le registre D0
    TRAP #15                         *arret du programme
IMPAIRE LEA MESSAGE3, A1              *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                    *Mettre 14 dans le registre D0
    TRAP #15                         *Affiche le message
    MOVE.B #9,D0                     *Mettre 14 dans le registre D0
    TRAP #15                         *arret du programme

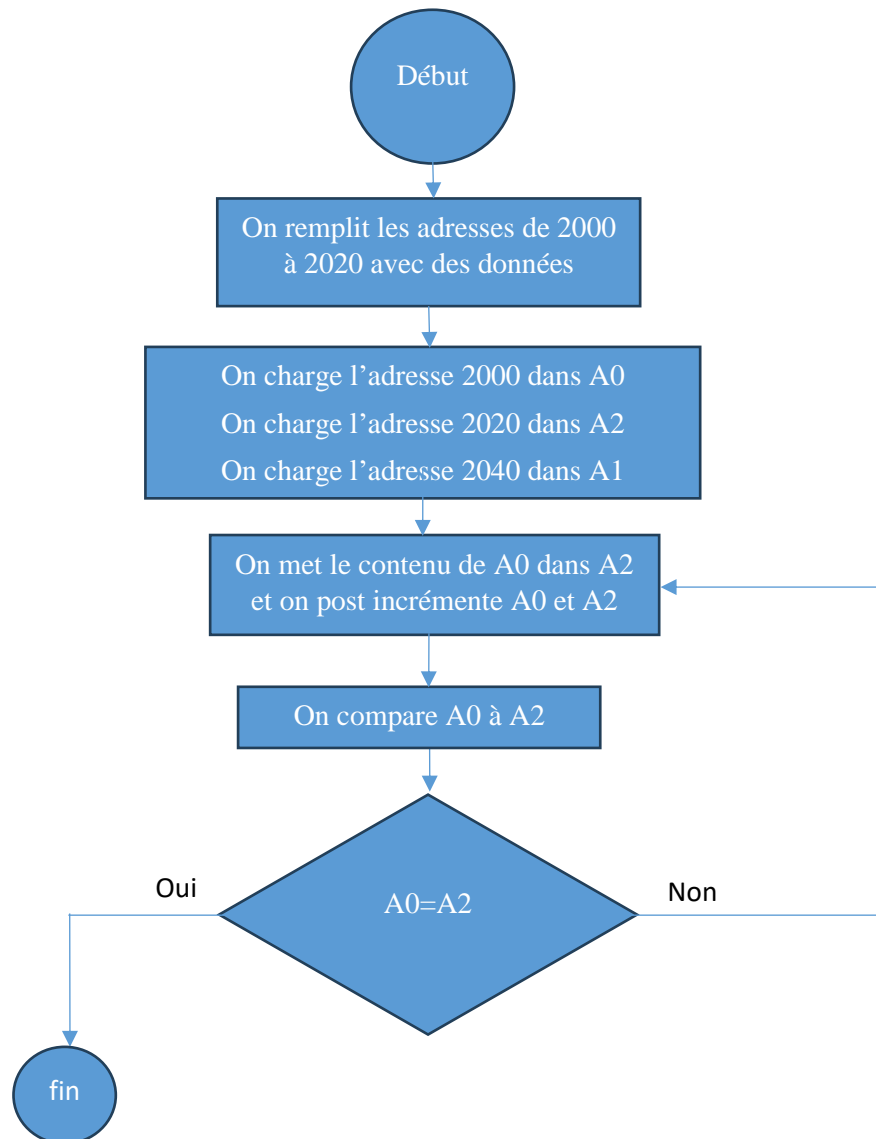
    SIMHALT                          ; halt simulator

* Put variables and constants here
MESSAGE0 DC.B '----- TP 2 EXO 3 -----',0,CR,LF
MESSAGE1 DC.B CR,LF,'Saisir un nombre: ',0
MESSAGE2 DC.B 'Le nombre est pair!',0
MESSAGE3 DC.B 'Le nombre est impaire!',0

    END      START                    ; last line of source
```

Exercice 4 :

Organigramme :



Programme :

```
*-----*
* Title      : TP2 EXO 4
* Written by : ERRARD - DACCACHE
* Date       : 13/03/24
* Description: EXO 4 a octet par octet
*-----*

    ORG      $1000
START:                                ; first instruction of program
CR EQU $0D                            *retour chariot (carriage return)
LF EQU $0A                            *Saut de ligne (Line Feed)
* Put program code here
    LEA MESSAGE0, A1                  *chargement de message dans l'adresse de
registre A1
    MOVE.B #14,D0                    *Mettre 14 dans le registre D0
    TRAP #15                          *Affiche le message
*-----*
    MOVE.B #1,$2000                  *on met 1 dans l'adresse 2000
    MOVE.B #2,$2020                  *on met 2 dans l'adresse 2020

    LEA $2000,A0                     *on charge le contenu de l'adresse 2000 dans A0
    LEA $2020,A2                     *on charge le contenu de l'adresse 2020 dans A2
    LEA $2040,A1                     *on charge le contenu de l'adresse 2040 dans A1
CYCLE MOVE.B (A0)+, (A1)+             *on met ce qui est dans l'adresse charger dans
A0 dans l'adresse charger dans A1 et on post incremente les 2 adresses
    CMPA A0,A2                       *on compare le contenu de A0 et A2
    BEQ FIN                          *si c'est le meme contenu on pass au code de
l'etiquette FIN
    BRA CYCLE                        *si le contenu n'est pas le meme on repasse au
code de l'etiquette CYCLE jusqu'a ce que le contenu soit le meme

FIN    MOVE.B #9,D0                  *on met 9 dans D0
    TRAP #15                          *on arrete le programme

    SIMHALT                          ; halt simulator

* Put variables and constants here
MESSAGE0 DC.B '----- TP 2 EXO 4 -----',0,CR,LF

    END      START                    ; last line of source
```

Pour la question b et c on a juste à changer dans la ligne 22 du code : « MOVE.B a MOVE.W pour le mot par mot ou MOVE.L pour mot long par mot long »

2. Si on change la destination initiale a 2041, on n'a pas de changement pour le code du bit par bit, vue qu'on fait une post-incrémentation de 1, mais pour le mot par mot et le mot long par mot long, si on fait on post-incrémentation de 1 on aura de la donnée qui va disparaître vu qu'on est entrain d'écrire sur une adresse qu'on a déjà utilisée (ex : si on fait un move.w sur l'adresse 2041, cela veut dire qu'on a écrit sur l'adresse 2041 et 2042, après une post-incrémentation de 1 on va réécrire sur l'adresse 2042 et on va écrire sur l'adresse 2043). Pour régler ce problème il faut qu'on effectue une post-incrémentation de 2 pour le mot par mot et une post-incrémentation de 4 pour les mot long par mot long.