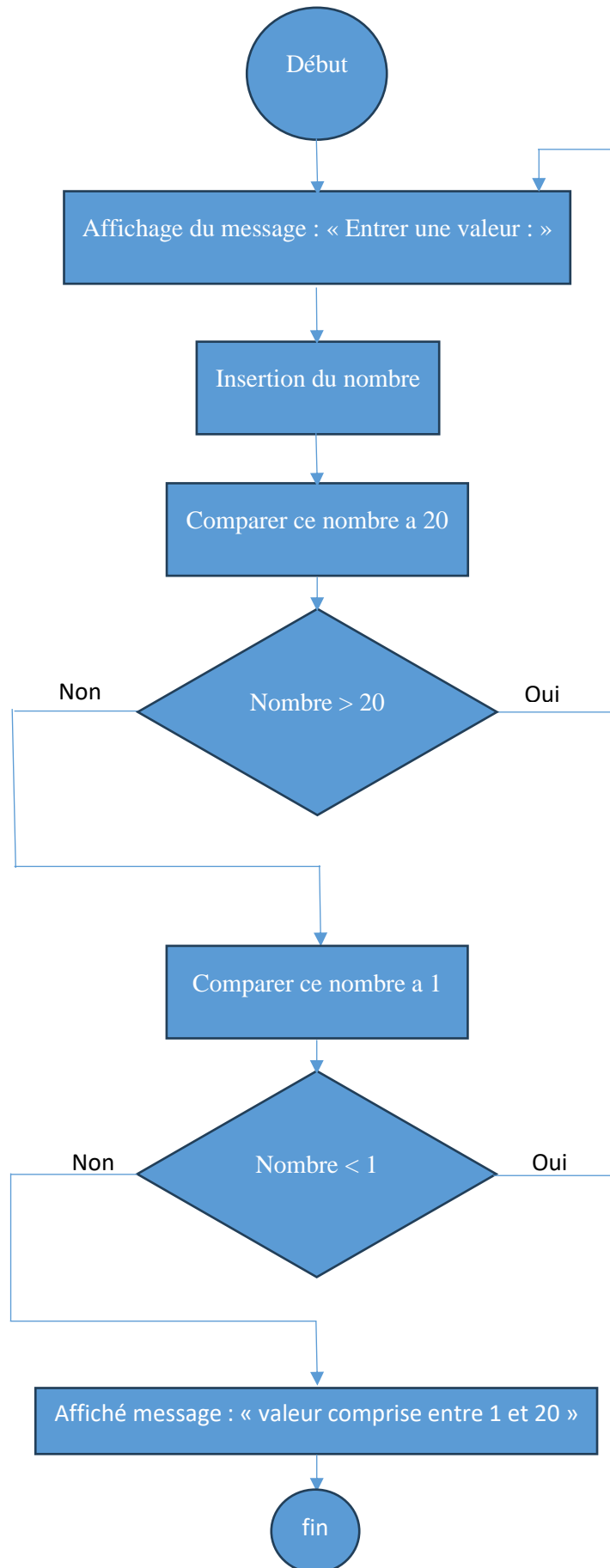


## Compte rendu TP 1

### Exercice 1 :

Organigramme :



Programme :

```
*-----*
* Title      : Exercice 1
* Written by : DACCACHE - ERRARD
* Date       : 06/03/24
* Description: T1
*-----*

    ORG      $1000
START:                                ; first instruction of program

*-----code for output-----*

LOOP    LEA MESSAGE, A1      *chargement de message dans l'adresse de
register A1
        MOVE.B #14,D0        *Mettre 14 dans le registre D0
        TRAP #15             *Affiche le message

*-----code for input-----*
        MOVE.B #4,D0
        TRAP #15             *Lecture d'un nombre du clavier vers D1.L
        CMP.L #20,D1         *Comparer la valeur entrée à 20
        BGT LOOP             *Si supérieur, on retourne à l'entrée
        CMP.L #1,D1          *Comparer la valeur entrée à 1
        BLT LOOP             *Si inférieur, on retourne à l'entrée
        LEA MESSAGE2, A1     *Affiche message2 si valeur entre 1 et 20
        MOVE.B #14,D0        *il faut avoir 14 dans D0 pour afficher la
chaine de caractere contenu dans A1
        TRAP #15             *execution de l'affichage du message

DONE    MOVE.B #9,D0         *il faut avoir 0 dans D0 pour arreter le
programme
        TRAP #15             *execution de l'arret du programme

CR EQU $0D                    *retour chariot (carriage return)
LF EQU $0A                    *Saut de ligne (Line Feed)

MESSAGE DC.B 'Entrer une valeur :',0
MESSAGE2 DC.B 'valeur comprise entre 1 et 20',0

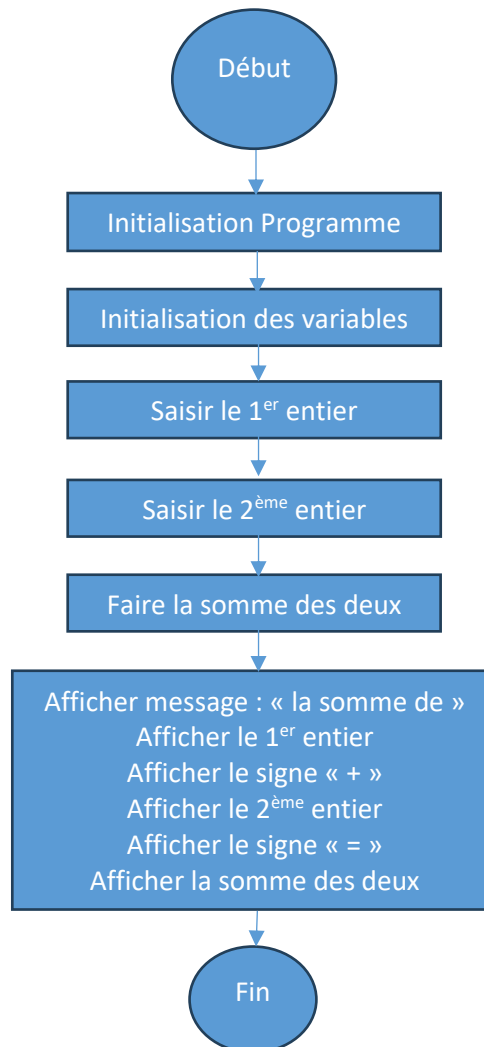
    SIMHALT                    ; halt simulator

* Put variables and constants here

    END      START            ; last line of source
```

**Exercice 2 :**

Organigramme :



Programme :

```
*-----*
* Title      : Exercice 2
* Written by : DACCACHE - ERRARD
* Date       : 06/03/24
* Description: TP1
*-----*

        ORG      $1000
START:                                ; first instruction of program

* Put program code here

CR EQU $0D                            *retour chariot (carriage return)
LF EQU $0A                            *Saut de ligne (Line Feed)

LOOP    LEA MESSAGE1, A1              *chargement de message1 dans l'adresse de
registre A1                          *
        MOVE.B #14,D0                *mettre 14 dans le registre D0
        TRAP #15                     *affiche du message1
        LEA MESSAGE2, A1              *chargement de message2 dans l'adresse de
registre A1                          *
        TRAP #15                     *affichage du message2

*-----premier entier-----*
        MOVE.B #4,D0                 *valeur du premier entier
        TRAP #15                     *affichage
        MOVE.B D1,D5                 *déplacement de D1 dans D5
        LEA MESSAGE3, A1              *chargement de message3 dans l'adresse de
registre A1                          *
        MOVE.B #14,D0                *mettre 14 dans D0
        TRAP #15                     *affichage du message3

*-----deuxième entier-----*
        MOVE.B #4,D0                 *valeur du deuxième entier
        TRAP #15                     *affichage
        MOVE.B D1,D6                 *déplacement de D1 dans D6
        ADD.B D5,D1                  *somme de D5 et D1
        LEA MESSAGE4, A1              *chargement de message4 dans l'adresse de
registre A1                          *
        MOVE.B #14,D0                *Mettre 14 dans le registre D0
        TRAP #15                     *affichage du message4
        MOVE.B D1,D7                 *déplacement de D1 dans D7
        MOVE.B D5,D1                 *déplacement du premier entier dans D1
        MOVE.B #3,D0                 **déplacement de 3 dans D0
        TRAP #15                     *affichage
        LEA plus, A1                  *chargement du caractere '=' dans l'adresse
de registre A1                        *
        MOVE.B #14, D0                *mettre 14 dans D0
        TRAP #15                     *affichage du caractere +
        MOVE.B D6,D1                 *déplacement du deuxième entier dans D1
        MOVE.B #3, D0                 *déplacement de 3 dans D0
        TRAP #15                     *affichage du deuxième entier
        LEA egal, A1                  *chargement du caractere '=' dans l'adresse
de registre A1                        *
        MOVE.B #14,D0                *mettre 14 dans D0
        TRAP #15                     *affichage du signe =
        MOVE.B D7, D1                 *on met la valeur du resultat qui est dans
D7 dans D1 pour l'afficher avec trap #15
        MOVE.B #3, D0                 *déplacement de 3 dans D0
        TRAP #15                     *affichage du resultat
```

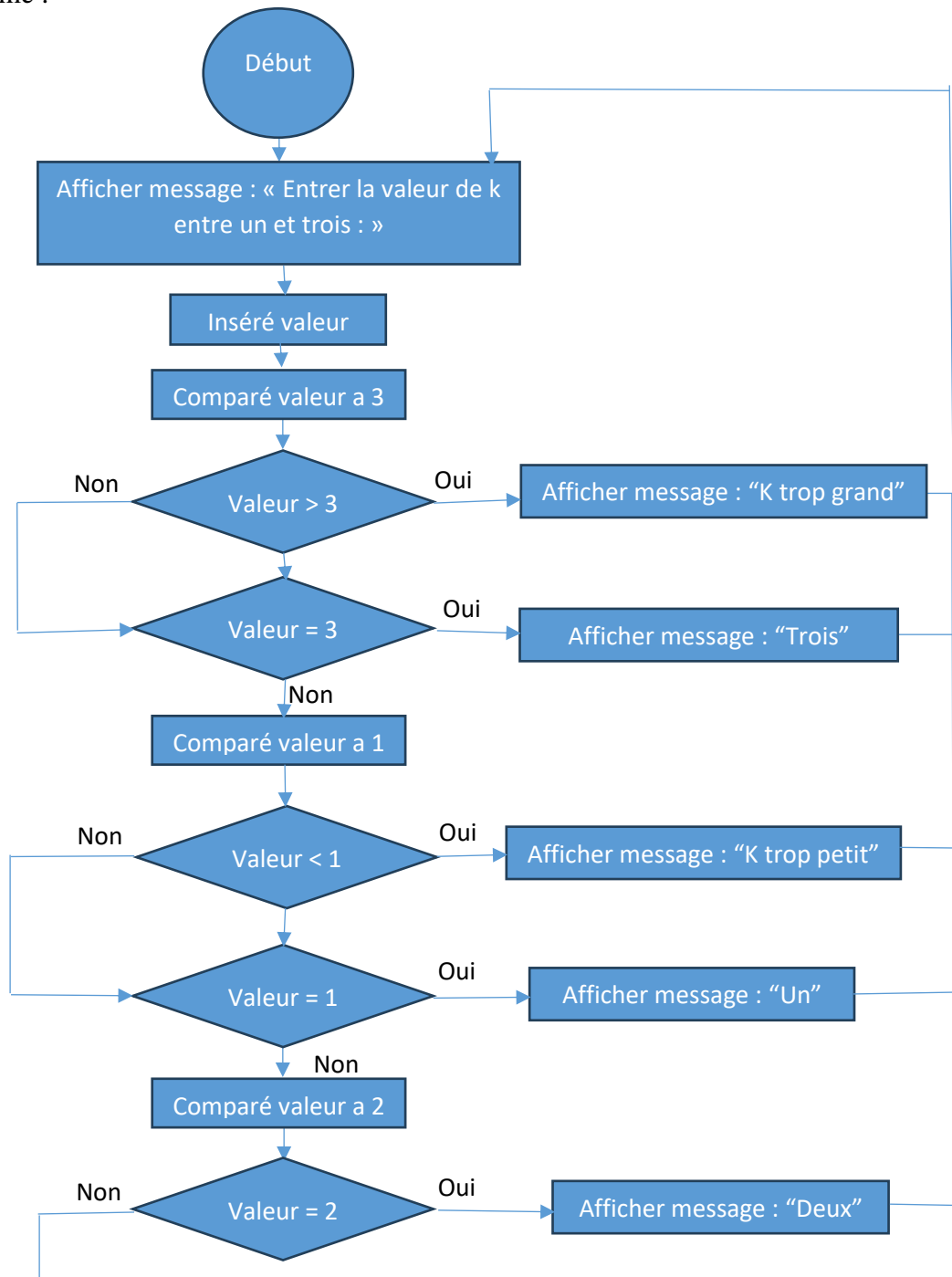
```
        SIMHALT                ; halt simulator

* Put variables and constants here
MESSAGE1 DC.B '----- TP1 EX02 -----',CR,LF,0
MESSAGE2 DC.B 'Donner le premier entier : ',0
MESSAGE3 DC.B 'Donner le deuxième entier : ',0,CR,LF
MESSAGE4 DC.B 'La somme de ',0
plus DC.B ' + ',0
egal DC.B ' = ',0

        END      START                ; last line of source
```

**Exercice 3 :**

Organigramme :



Programme :

```
*-----
* Title : EXERCICE 3
* Written by : ERRARD - DACCACHE
* Date : 06/03/24
* Description: TP1
*-----

START: ORG 1000 ; first instruction of program
* Put program code here

BOUCLE LEA message1, A1      *Chargement du message1 dans A1
      move.b #14,D0          *on met 14 dans D0 pour afficher une chaine de
caractere
      TRAP #15               *Afficher le message1
      LEA message2,A1        *Chargement du message2 dans A1
      TRAP #15               *Afficher le message2
      move.b #4,D0           *on met 4 dans D0 pour lire une entree du
clavier apres la touche enter
      TRAP #15               *Lecture d'un entier
      CMP.L #3,D1            *Comparaison de l'entier avec 3
      BGT SUP                *Si supérieur a 3, aller à l'étiquette SUP
      CMP.L #1,D1            *Comparaison de l'entier avec 1
      BLT INF                *Si inférieur a 1, aller à l'étiquette INF
      CMP.L #1,D1            *Comparaison de l'entier avec 1
      BEQ UN                 *Si égal à 1, aller à l'étiquette UN
      CMP.L #2,D1            *Comparaison de l'entier avec 2
      BEQ DEUX               *Si égal à 2, aller à l'étiquette DEUX
      CMP.L #3,D1            *Comparaison de l'entier avec 3
      BEQ TROIS              *Si égal à 3, aller à l'étiquette TROIS
      BRA BOUCLE             *et finalement aller a l'etiquette boucle pour
reprendre le programme
SUP LEA superieur,A1         *Charger le message "k trop grand" dans A1
  move.b #14,D0              *on met 14 dans D0 pour afficher une chaine de
caractere
  TRAP #15                   *Afficher le message "k trop grand"
  BRA BOUCLE                 *retour a l'etiquette BOUCLE
INF LEA inferieur,A1         *Charger le message "k trop petit" dans A1
  move.b #14,D0              *on met 14 dans D0 pour afficher une chaine de
caractere
  TRAP #15                   *Afficher le message "k trop petit"
  BRA BOUCLE                 *retour a l'etiquette boucle
UN LEA nombre_un,A1          *Chargement de la chaine de caractere
nombre_un dans A1
  move.b #14,D0              *on met 14 dans D0 pour afficher une chaine de
caractere
  TRAP #15                   *Afficher le message "un"
  BRA BOUCLE                 *retour a l'etiquette boucle
DEUX LEA nombre_deux,A1     *Chargement de la chaine de caractere
nombre_deux dans A1
  move.b #14,D0              *on met 14 dans D0 pour afficher une chaine de
caractere
  TRAP #15                   *Afficher le message "deux"
  BRA BOUCLE                 *retour a l'etiquette BOUCLE
TROIS LEA nombre_trois,A1   *Chargement de la chaine de caractere
nombre_trois dans A1
  move.b #14,D0              *on met 14 dans D0 pour afficher une chaine de
caractere
  TRAP #15                   *Afficher le message "trois"
  BRA BOUCLE                 *retour a l'etiquette BOUCLE
```

```
                SIMHALT                ; halt simulator

* Put variables and constants here
CR EQU $0D
LF EQU $0A
message1 dc.b CR,LF,'----- TP1 Exo3 -----',CR,LF,0
message2 dc.b 'Entrer la valeur de k entre un et trois: ',0
superieur dc.b CR,LF,'k trop grand',0
inferieur dc.b CR,LF,'k trop petit',0
nombre_un dc.b CR,LF,' un ',0
nombre_deux dc.b CR,LF,'deux',0
nombre_trois dc.b CR,LF,'trois',0

                END                START                ; last line of source
```