

### **Devoir 3 sur l'assembleur Structure des ordinateurs 8INF 212**

Soit le programme de la page 2

- Écrivez-le tel qu'il est.

**Question 1:** Corrigez ce programme aux endroits indiqués pour pouvoir obtenir comme résultat la figure de la page 3. Expliquez brièvement pourquoi les lignes que vous avez corrigées ne fonctionnent pas. En l'exécutant, vous obtiendrez le résultat tel qu'il est indiqué en page 3.

**Question 2:** Modifiez le programme obtenu à la question 1 comme suit:

- a) Demandez à l'utilisateur de saisir, à partir du clavier, deux valeurs non négatives A et B;
- b) Afficher la valeur A dans la colonne 1;
- c) Afficher le signe + ou - dans la colonne 2;
- d) Afficher la valeur B dans la colonne 3;
- e) Afficher le signe = dans la colonne 4;
- f) Afficher le résultat de la somme et de la soustraction de A et B dans la colonne 5;

1. On suppose que les données et le résultat tiennent sur un seul chiffre.
2. On suppose que les données et le résultat tiennent sur plus d'un chiffre.

Une trace d'exécution du programme demandé est fournie en page 4.

#### **Travail demandé:**

1. rapport écrit expliquant votre travail.
2. une disquette contenant le source de la question 1, le source de la question 2.1 et de la question 2.2 (dans des fichiers différents). Il est recommandé de commenter le plus possible vos programmes.

```

TITLE  devoir 3
SPILE  SEGMENT STACK
      DW 100 DUP (?)
SPILE  ENDS

```

```

SDATA SEGMENT

```

```

ligne_0      db  'Ce TP est intéressant pour débiter la programmation en assembleur ', '$'
ligne_1      db  '
              db  '|          |          |          |          |          |          |          |'
              db  '|          |          |          |          |          |          |          |'
ligne_centre db  '|          |          |          |          |          |          |          |'
ligne_fin    db  '|          |          |          |          |          |          |          |'
SDATA ENDS

```

```

SCODE SEGMENT

```

```

      ASSUME CS:SCODE, DS:SDATA ; génération des adresses des segments de code et de données

```

```

DEBUT:

```

```

      ; Initialiser le registre DS
      MOV AX, SDATA
      MOV DS, AX

```

```

      MOV DX, OFFSET ligne_0
      MOV AH, 09h
      INT 21h

```

```

      jmp debut      ; <---- 1er bug

```

```

      MOV DX, OFFSET ligne_1
      MOV AH, 09h
      INT 21h

```

```

      MOV DX, OFFSET ligne_centre

```

```

boucle: MOV AH, 09h
        INT 21h

```

```

      dec dx      ; <----- 2ème bug

```

```

      mov ah,01h
      int 16h    ; <----- 3ème bug

```

```

      jz boucle   ; <---- 4ème bug

```

```

      MOV DX, OFFSET ligne_fin
      MOV AH, 09h
      INT 21h

```

```

      ; Terminer le programme
      MOV Ax, 4C00h
      INT 21h

```

```

SCODE ENDS

```

```

      END DEBUT

```

**Après correction et ajout de quelques lignes de code, on doit obtenir ce qui suit :**

Vous entrez dans le monde merveilleux de l'assembleur!

colonne 1	colonne 2	Colonne 3	colonne 4	colonne 5
-----------	-----------	-----------	-----------	-----------

Appuyer uniquement sur ESCAPE pour sortir du programme ... à la prochaine

## **1. Après rajout d'autres lignes de code, on obtient ce qui suit :**

Veillez taper la première valeur

4

Veillez taper la deuxième valeur

2

Vous entrez dans le monde merveilleux de l'assembleur!

colonne 1	colonne 2	colonne 3	colonne 4	colonne 5
4	+	2	=	6
2	-	4	=	-2

Appuyer uniquement sur ESCAPE pour sortir du programme ... à la prochaine !

---

Veillez taper la première valeur

5

Veillez taper la deuxième valeur

6

Vous entrez dans le monde merveilleux de l'assembleur!

colonne 1	colonne 2	colonne 3	colonne 4	colonne 5
5	+	4	=	9
5	-	4	=	-1

Appuyer uniquement sur ESCAPE pour sortir du programme ... à la prochaine !

---

## **2. Après rajout d'autres lignes de code, on obtient ce qui suit :**

Veillez taper la première valeur

14

Veillez taper la deuxième valeur

12

Vous entrez dans le monde merveilleux de l'assembleur!

colonne 1	colonne 2	colonne 3	colonne 4	colonne 5
14	+	12	=	26
14	-	12	=	2

Appuyer uniquement sur ESCAPE pour sortir du programme ... à la prochaine !