

TP n° 2 : Conditionnelles ; premières boucles



Ce sujet comporte deux parties :

- vous pouvez traiter la première partie de ce sujet en utilisant les éléments vus dans les deux premières parties du sujet de TD n° 4 ;
- pour les boucles, travaillez préalablement la troisième partie de l'énoncé de TD.

1 Conditionnelles

1. [intervalle.asm]

Ecrivez un programme assembleur demandant à l'utilisateur de saisir deux valeurs a et b , puis une troisième valeur x .

(a) On suppose que $a < b$. Le programme affiche ensuite :

- 0 si $x < a$
- 1 si $x \in [a; b]$
- 2 si $x > b$

(b) Que se passe-t-il si $a = b$? si $a > b$?

Corrigez le programme pour assurer d'avoir $a \leq b$.

2. [calculette.asm]

Le programme que vous devrez écrire simule le comportement d'une calculatrice, basée sur la structure algorithmique du Cas Parmi : il demande à l'utilisateur de saisir deux valeurs a et b , puis une valeur c ; ensuite :

- si $c = 1$, il affiche la valeur de $a + b$
- si $c = 2$, il affiche la valeur de $a - b$
- si $c = 3$, il affiche la valeur de $a \times b$
- si $c = 4$, il affiche la valeur de a / b
- si $c = 5$, il affiche la valeur de $a \% b$ (calcul du reste)

(a) Ecrivez le programme, en supposant que l'utilisateur saisira forcément des valeurs correctes : valeur de c comprise entre 1 et 5; valeur de b non nulle pour les deux opérations de division.

(b) Que se passe-t-il si la valeur de c n'est pas correcte? **(comme si on avait $c=1$)**
Corrigez, en n'affichant rien si $c = 0$, et en calculant la somme dans le cas contraire.

(c) Corrigez le programme pour afficher 0 dans le cas où $b = 0$ avec 4 ou 5 comme valeur de c .
... afficher 0 dans le cas où [$b=0$ ET ($c=4$ OU $c=5$)]

2 ... et boucles

1. *Faire ... TantQue* : [calculette.asm] (suite)

Pour un couple de valeurs a et b donné, on souhaite pouvoir tester plusieurs opérations :

- pour une valeur cohérente de c on procède comme précédemment
- lorsque l'utilisateur saisit une valeur de c au moins égale à 6 on ne fait rien
- le programme s'arrête lorsque l'utilisateur saisit la valeur 0 pour c .

Modifiez votre programme en conséquence.

2. *TantQue ... Faire* : [pgcd.asm]

On veut écrire un programme en assembleur permettant de calculer le PGCD de deux nombres a et b :

- la première partie du programme consiste à saisir deux valeurs strictement positives pour a et b (même si cela nécessite de faire plusieurs essais...)
- ensuite on fait en sorte que la plus grande des deux valeurs soit dans a , en les échangeant au besoin
- puis on calcule le PGCD en utilisant l'algorithme d'Euclide¹ :

```

reste ← a % b
TantQue reste ≠ 0 Faire
    a ← b
    b ← reste
reste ← a % b
FinTantQue
{ le PGCD est le dernier reste non nul : }
{ il est donc contenu dans la variable b }

```

0

$$\begin{array}{r|l} 306 & 24 \\ 66 & 12 \\ 18 & \end{array}$$

1

$$\begin{array}{r|l} 24 & 18 \\ 6 & 1 \end{array}$$

2

$$\begin{array}{r|l} 18 & 6 \\ 0 & 3 \end{array}$$

- finalement on affiche la valeur obtenue pour le PGCD.

Ecrivez le programme pour ce traitement.

NB : vous pouvez vérifier qu'il n'est pas nécessaire d'ordonner les valeurs de a et b : essayez votre programme après avoir mis en commentaire la partie du code qui ordonne les valeurs de a et b (pensez à renuméroter les instructions, pour que les sauts se fassent au bon endroit).

3 Nommage des fichiers / de l'archive

Si vous souhaitez avoir un retour sur votre travail :



- nommez vos fichiers tel qu'indiqué dans l'énoncé
- renommez votre répertoire avec vos nom et prénom (sans espace ni accent),
COMME CI-DESSOUS
⇒ TP1_JAILLET_Christophe
- si vous avez travaillé à deux ou plus, nommez le répertoire avec tous vos noms
⇒ TP1_DEBARLE_Noel_JAILLET_Christophe_MOUSE_Mickey
- compressez l'archive au format zip

1. calcul du PGCD par divisions successives : il s'agit d'une adaptation de la méthode des différences, en utilisant la division euclidienne