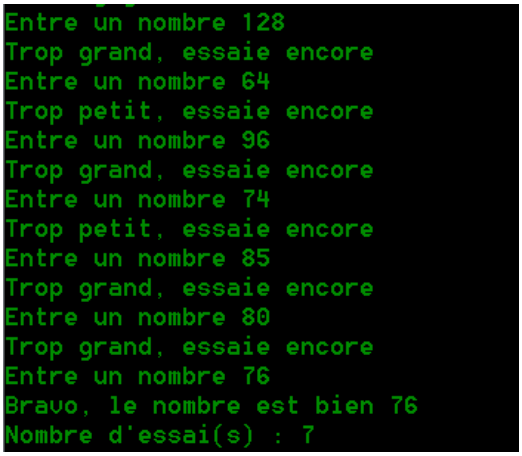
Semaine 7 – Langage machine – Exercices

Exercice n°7.1 – Programme à réaliser en NASM 32 bits

En utilisant l'IDE SASM, écrivez un programme en NASM qui demande à l'utilisateur d'entrer un nombre entier en décimal non signé jusqu'à ce qu'il trouve le nombre mystère.

Le nombre mystère est codé en mémoire en dur dans le programme, initialisé en hexadécimal dans la section .data.

Travaillez avec des nombres qui ont une taille de 1 octet.

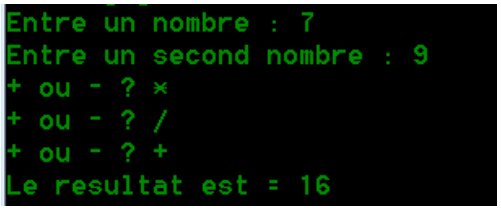


Exercice n°7.2 – Programme à réaliser en NASM 32 bits

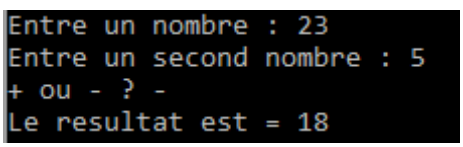
En utilisant l'IDE SASM, écrivez un programme en NASM qui demande à l'utilisateur d'entrer un nombre entier en décimal non signé, un second nombre entier en décimal non signé, et ensuite un opérateur (+ ou -).

Travaillez avec des nombres qui ont une taille de *double word* (4 octets).

L'utilisateur doit entrer un opérateur (+ ou -) correctement, vous devez lui redemander d'entrer un opérateur (+ ou -) jusqu'à ce que ce soit correct.



Votre programme affiche ensuite le résultat de l'addition ou de la soustraction des deux nombres.



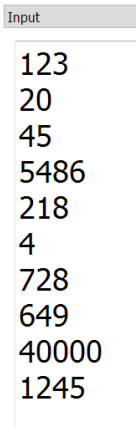
Exercice n°7.3 – Programme à réaliser en NASM 32 bits

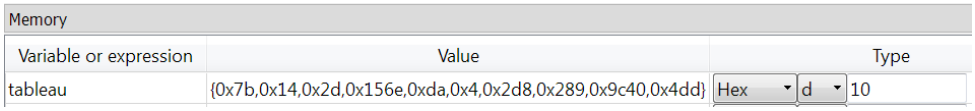
En utilisant l'IDE SASM, écrivez un programme en NASM qui demande à l'utilisateur d'entrer 10 nombres entiers en décimal non signé.

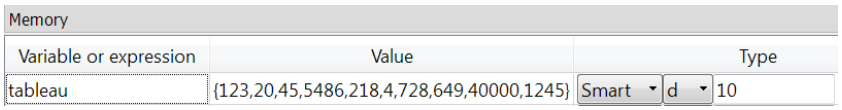
Vous devez stocker ces 10 nombres en mémoire dans un tableau de *double word*. Utiliser la directive resd dans la section .bss pour ce faire.

Par exemple,

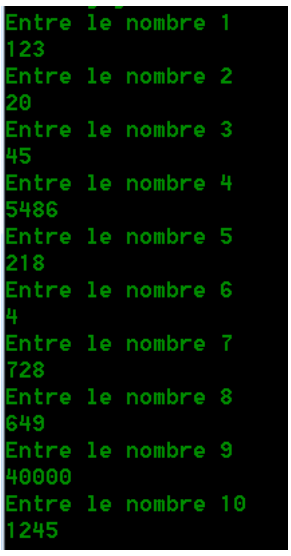
avec les *inputs* ci-dessous,
vous devez pouvoir visualiser en mémoire le tableau en mode *Hex* ou *Smart* :







L'exécution en fenêtre de commandes Windows donne, selon le même exemple que dans SASM, ceci :



Ensuite, en parcourant ce tableau en mémoire, trouvez le nombre le plus grand et le plus petit et affichez-les à l'écran.

L'exemple précité donne dans SASM :