

Chapitre 3

Ordinapoche

Mémo

code	abréviation	code	abréviation
0	INP	5	SUB
1	OUT	6	SHT
2	CLA	7	JMP
3	STO	8	TAC
4	ADD	9	HRS

3.1 Echange du contenu de deux variables

Ecrire le programme qui échange deux valeurs déjà présentes en mémoire centrale, d'adresse respective 20 et 21 (donc on *ne* demande *pas* d'affichage des contenus).

3.2 Addition de deux entiers saisis au clavier

1. Ecrire un programme qui demande à l'utilisateur de saisir deux nombres, que l'on supposera compris entre 0 et 499, qui les additionne et qui affiche le résultat.
2. Si l'utilisateur saisit deux nombres dont la somme est supérieure à 999, que se passe t'il ?

3.3 Un compteur infini

1. Commenter chaque ligne du programme ci-après.
2. Que fait ce programme ?
3. Quelle est l'adresse de la variable `nb` ?
4. L'identificateur `un` correspond-il à une constante ou une variable ?
5. Peut-on se passer de l'instruction `STO` ? Quel problème va survenir ?

Adresse	Contenu	Commentaires
00	CLA un	
01	STO nb	
02	OUT nb	
03	ADD un	
04	JMP 01	
05	1	un
06	?	nb

3.4 De l'usage de l'instruction SHIFT

En utilisant l'instruction **SHT**, écrire un programme qui saisit au clavier un seul nombre n de 2 chiffres ($10 \leq n \leq 99$), qui le décompose en dizaine et unité, et qui affiche sa décomposition.

3.5 Compte à rebours pour le lancement d'une fusée

1. Écrire un programme qui se charge de l'affichage du compte à rebours pour le lancement d'une fusée. Le compte à rebours démarre à 10 et se termine à 0. Utiliser obligatoirement une instruction de test.
2. Écrire un programme qui simule un compteur qui démarre à 0 et qui affiche tous les entiers jusqu'à 10.

Remarque : l'ordinapoche dispose de **TAC** comme *unique* instruction conditionnelle, or celle-ci ne fait que vérifier si le registre **ACC** est nul ou non ; cette limitation n'empêche pas de vérifier qu'une variable atteint une valeur donnée.

3.6 Nombre secret

1. Écrire un programme qui vous propose de deviner ce nombre, en indiquant par 0 ou par 1 si le nombre que vous entrez est trop petit ou trop grand. Lorsque le nombre est trouvé, il s'affiche sur le périphérique de sortie.
2. Quelle stratégie devez-vous adopter pour trouver ce nombre en un minimum d'essais ?
3. Modifier le programme de telle sorte que le nombre s'affiche ainsi que le nombre de coups qu'il a été nécessaire pour le trouver.

3.7 Somme d'une suite par modification d'une instruction

On suppose dix nombres, tous inférieurs à 20, mémorisés de l'adresse 50 à l'adresse 59.

En jouant sur le fait que l'ordinapoche est incapable de distinguer instruction et donnée, on se propose de mettre en place une structure itérative qui va permettre de sommer ces dix nombres.

Le programme à écrire se décompose de la façon suivante : additionner l'un des dix nombres, transformer l'instruction d'addition de telle sorte qu'elle permette l'addition du nombre suivant en mémoire, gérer un compteur qui indique quand s'arrêter et enfin afficher le résultat.