

I12 - TD 3

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

Exercice 1. 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).

Exercice 2. 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. Si l'utilisateur saisit deux nombres dont la somme est supérieure à 999, que se passe t'il ?

Exercice 3. Commenter chaque ligne du programme ci-dessous. Que fait ce programme? Quelle est l'adresse de la variable `nb`? L'identificateur `un` correspond-il à une constante ou une variable? 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

Exercice 4. 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.

Exercice 5.

1. Ecrire 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. Ecrire 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; comment dans ces conditions vérifier qu'une variable atteint une valeur donnée ?

Exercice 6. On suppose un nombre inconnu à l'adresse 50.

1. Ecrire 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 qui vous ont été nécessaire pour le trouver.

Exercice 7. 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.

Correction

Exercice 1. Echange du contenu de deux variables.

Adresse	Instruction	Commentaire
00	CLA a	
01	STO tmp	sauvegarde de a
02	CLA b	
03	STO a	a prend la valeur de b
04	CLA tmp	
05	STO b	b prend la valeur de a
06	HRS	fin de programme
07	?	donnée a (variable)
08	?	donnée b (variable)
09	?	donnée tmp (variable)

Exercice 2. Addition de deux entiers saisis au clavier.

Adresse	Instruction	Commentaire
00	INP a	saisi du 1er nombre
01	INP b	saisi du 2nd nombre
02	CLA a	initialisation ACC
03	ADD b	addition de b
04	STO c	mémorisation du résultat
05	OUT c	affichage
06	HRS	fin de programme
07	?	donnée a (variable)
08	?	donnée b (variable)
09	?	donnée c (variable)

Exercice 3. De l'usage de l'instruction SHIFT.

Adresse	Instruction	Commentaire
00	INP n	saisi d'un nombre à deux chiffres
01	CLA n	
02	SHT 12	ACC contient le chiffre des dizaines
03	STO d	stockage du chiffre des dizaines
04	CLA n	
05	SHT 22	ACC contient le chiffre des unités
06	STO u	stockage du chiffre des unités
07	OUT d	affichage du chiffre des dizaines
08	OUT u	affichage du chiffre des unités
09	HRS	fin de programme
10	?	donnée n (variable)
11	?	donnée d (variable)
12	?	donnée u (variable)

Exercice 4. Un compteur infini.

Adresse	Contenu	Commentaires
00	CLA un	mise à zéro d'ACC et addition du contenu d'adresse 05 (1)
01	STO nb	emmagasiner le contenu de l'ACC à l'adresse 06
02	OUT nb	imprimer le contenu d'adresse 06
03	ADD UN	additionner à l'ACC le contenu d'adresse 05
04	JMP 01	saut inconditionnel à l'instruction d'adresse 01
05	1	donnée UN (constante)
06	?	donnée nb (variable)

Exercice 5. Compte à rebours pour le lancement d'une fusée.

Adresse	Instruction	Commentaire
00	CLA nb	
01	SUB UN	soustraire 1
02	STO nb	emmagasiner ACC à l'adresse 07
03	OUT nb	imprimer le contenu d'adresse 07
04	TAC 00	si ACC \neq 0 alors saut à l'adresse 00
05	HRS	arrêter et remettre à l'état initial
06	1	donnée UN (constante)
07	11	donnée nb (variable initialisée par programme)

Adresse	Instruction	Commentaire
00	OUT nb	imprimer le contenu d'adresse 07
01	CLA nb	
02	ADD UN	ajouter 1
03	STO nb	emmagasiner ACC à l'adresse 07
04	CLA cpt	gestion de la borne d'arrêt
05	SUB UN	
06	STO cpt	
07	TAC 00	si ACC \neq 0 alors saut à l'adresse 00
08	HRS	arrêter et remettre à l'état initial
09	1	donnée UN (constante)
10	0	donnée nb (variable initialisée par programme)
11	11	donnée cpt (variable initialisée par programme)

Exercice 6. Nombre secret

Adresse	Instruction	Commentaire
00	INP prop	prop(osition)
01	CLA sec	copie de sec(ret)
02	STO bis	
03	CLA bis	$bis \leftarrow bis - 1$
04	SUB UN	
05	STO bis	
06	TAC 15	
07	CLA prop	$bis = 0, prop \leftarrow prop - 1$
08	SUB UN	
09	STO prop	
10	TAC 13	
11	OUT sec	trouve $prop = 0$
12	HRS	
13	OUT UN	trop grand
14	JMP 00	
15	CLA prop	$bis \neq 0$
16	SUB UN	
17	STO prop	
18	TAC 03	
19	OUT ZERO	trop petit
20	JMP 00	

Pour le nombre de coups, on intercale la gestion d'un compteur (CLA cpt ; ADD UN ; STO cpt), initialisé à 0, entre les lignes 02 et 03. On intercale de même l'affichage du nombre de coups (OUT cpt) entre les lignes 11 et 12.

Exercice 7. Somme d'une suite de nombres par modification d'une instruction.

Adresse	Instruction	Commentaire
00	CLA res	variable res contenant la somme
01	ADD 50	ajout d'un nombre de la suite
02	STO res	
03	CLA 01	modification de l'instruction ADD 50
04	ADD UN	
05	STO 01	
06	CLA cpt	compteur de boucle
07	SUB UN	
08	STO cpt	
09	TAC 00	
10	OUT res	affichage de la somme
11	HRS	
12	1	donnée UN (constante)
13	10	donnée cpt (variable)
14	0	donnée res (variable)