TP N° 4

Exercice 1 :

Ecrire un programme pour :

1. Déclarer deux tableaux de 100 mots, tab1 et tab2.
2. Remplir le premier avec les 100 premiers nombres positifs multiple de 2 et le second avec les 100 premiers nombres positifs multiple de 3.
3. Empiler dans la pile les éléments qui sont communs aux deux tableaux
4. Additionner les éléments de la pile en les dépilant et stocker le résultat à l'adresse hexadécimale FAD
5. Après l'empilement des éléments dans la pile, quelle est l'adresse de son sommet ?

Exercice 2 :

Ecrire un programme, en langage assembleur 8086, qui permet de saisir deux nombres sur 8 bits au clavier (entre 0 et 255), et aussi l'opération à effectuer (+ , - , * et /). Ensuite calculer les résultats dans l'ensemble des entiers naturels et afficher les résultats à l'écran. Le programme doit gérer les exceptions et les erreurs suivant en affichant des messages claires du problème :

1. La saisie d'un caractère qui n'est pas un chiffre,
2. La saisie d'un nombre qui n'est pas dans l'intervalle 0…255,
3. La saisie d'un caractère d'opération invalide,
4. La division par 0,
5. Un résultat de soustraction négatif.

Les vérifications des trois premiers cas doivent être effectuées jusqu'à que les opérandes et les opérations soient valide.

Le codage ASCII

La table suivante donne le code ASCII des caractères usuels. L'abscisse indique le chiffre, l'ordonnée la dizaine (en hexadécimal). Ainsi, le code ASCII du caractère + est 2Bh.

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
2.		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	