Université Akli Mohand Oulhadj-Bouira	*Année universitaire:2022-2023*
Faculté des sciences et sciences appliquées	*Module: Architecture des ordinateurs*
Département d'Informatique	*TD N°3*

EXERCICE N°1:

Donner un programme assembleur x86 pour calculer la factorielle d'un entier en utilisant un sous programme.

- lire l'entier à partir le clavier.
- passage des paramètres par **pile**.
- retour de résultat par **pile**.
- sauvegarder le résultat dans la case 100h.

EXERCICE N°2:

Donner un programme assembleur x86 pour calculer la factorielle d'un entier en utilisant un sous programme.

- lire l'entier à partir de l'adresse FF00h: 0100h.
- passage des paramètres par **registres**.
- retour de résultat par **pile**.
- afficher le résultat sur l'écran.

EXERCICE N°3:

Soit le jeu d'instruction suivant:

انقل ع1، ع2	ع2:=ع1، ع قد يكون سجل، عدد أو خانة في الذاكرة
اجمع س1،س2،س3	س3:=س1+ س2
اطرح س1،س2،س3	س3:=س1- س2
اضرب س1،س2،س3	س3:=س1* س2
اقسم س1،س2،س3	س3:=الحاصل، س2:=الباقي
قارن س1،س2	طرح شكلي
اقفز عنوان	ع:= عنوان
اقفزصفر عنوان	إذا النتيجة 0، ع:= عنوان
اقفزأقل عنوان	بعد قارن، إذا الأول أصغر،ع:= عنوان
اقفزمختلف عنوان	بعد قارن، إذا مختلفان،ع:= عنوان
نهاية	نهاية البرنامج

1- Donner un programme pour faire les opérations suivantes sur 03 nombres x,y et z stockés dans [100h], [104h] et [108h] respectivement. Ensuite, calculer:
x:=x+1;
y:=y-1;
y:=y*x;
z:=z/y;
Sauvegarder le résultat de la dernière opération dans [200h] et [204h].
2- Définir pour ce processeur: la taille mémoire adressable, le mot mémoire, et l'adresse de la dernière case. es qu'on a besoin de segmenter la mémoire? justifier.

EXERCICE N°4:

Soient le jeu d'instructions suivants:

- **LOAD [adr]** : La valeur de l'emplacement mémoire pointé par l'adresse est copiée dans l'accumulateur.
- **SAVE [adr]:** La valeur de l'accumulateur est copiée à l'emplacement pointé par l'adresse.
- **ADD [adr]:** La valeur de l'emplacement pointé par l'adresse est ajoutée à la valeur de l'accumulateur.
- **SUB [adr]:** La valeur de l'emplacement pointé par l'adresse est soustraite à la valeur de l'accumulateur.
- **INC [adr]:** La valeur de l'emplacement pointé par l'adresse est incrémentée.
- **DEC** [adr]: La valeur de l'emplacement pointé par l'adresse est décrémentée.
- **NULL [adr]:** La valeur de l'emplacement pointé par l'adresse est mise à zéro.
- **TST [adr]:** l'instruction suivante est ignorée si et seulement si la valeur de l'emplacement pointé par l'adresse est nulle.
- **JMP etq:** Le programme se poursuit à l'adresse indiquée par etq dans le programme.
- **JL [adr];** ignorer l'instruction suivante si ACC < [adr]
- **FIN** : L'exécution du programme est terminée.

En se basant sur le jeu d'instruction précédent, donner les programmes pour:
1- Calculer: [300h]:= [100h] * [200h]
2- Calculer: [300h]:= [100h] / [200h]

EXERCICE N°5:

Soit le jeu d'instructions d'un processeur suivant:

Instruction ;	***Signification***
stop;	*Fin*
saut i ;	*compteur de programme ← i*
saut Ri j;	*Si la valeur Ri = 0, compteur de programme ← j*
val x Ri ;	*Ri ← valeur de x*
lect i Rj ;	*Rj ← contenu d'adresse i*
ecrit Ri j;	*Ri → dans la mémoire d'adresse j*
add Ri Rj ;	*Rj← Ri + Rj*
soustr Ri Rj ;	*Rj← Ri - R j*
mult Ri Rj;	*Rj← Ri * R j*
div Ri Rj;	*Rj←Ri / R j (division entière)*

En se basant sur le jeu d'instruction précédant, écrire les programmes en assembleur correspondant aux cas suivants:
1. Tester un nombre A s'il est pair on met 1 dans le registre R5 sinon 0.
2. Calculer la somme des nombres naturels de p à q inclus (où p < q).