Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira -
ⵜⴰⵙⴷⴰⵡⵉⵜ ⴰⴽⵍⵉ ⵎⵓⵃⴻⵏⴷ ⵓⵍⵃⴰⵊ - ⵜⵓⴱⵉⵔⴻⵜ -

وزارة التعليم العالي والبحث العلمي
جامعة أكلي محند أولحاج
- البويرة -

Département d'informatique **Année: 2021/2022**

Module: Architecture des ordinateurs **Examen: L2/S3 (1h+20m)**

Exercice n°1(05 points)**:** Donner un programme assembleur 8086 qui demande d'entrer un entier ***n*** par le clavier, voir s'il est ***pair*** ou ***impair***, afficher le résultat sur l'écran.

```
    mov dx, offset msg
    mov ah, 9   ;afficher msg
    int 21h
    ; read character in al:
    mov ah, 1
    int 21h
    mov bl,2
    div bl
    cmp ah,0
    jz paire
    mov ah,02
    mov dl,13
    int 21h ; ; saut de la ligne
    mov dx, offset msgImpair
    mov ah, 9   ;afficher msg
    int 21h
    jmp fin
paire:
    mov ah,02
    mov dl,13
    int 21h ; saut de la ligne
    mov dx, offset msgPair
    mov ah, 9   ;afficher msg
    int 21h
 fin:ret  ; stop

    msg db "donner un entier:  $"
    msgPair db "Le nombre est paire $";
    msgImpair db "Le nombre est impaire$";
```

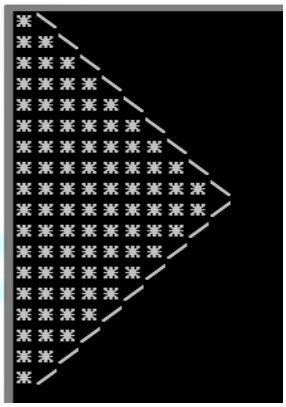
Exercice n°2(05 points)**:** Donner un programme assembleur 8086 pour calculer le factoriel d'un entier stocké dans la case [300h]. Utiliser une fonction nommée ***fact*** située dans ***le même segment*** que le programme principale. Le passage de l'entier

doit se fait par la pile. Le retour de résultat doit se fait par ***registre***. Stocker le résultat à partir de l'adresse 400h. Afficher le résultat sur l'écran.

```
  mov [300h],10; n=10 for example
  Mov al,[300h]
  Mov ah,00
  Push ax
  Call fact
  mov [400h],ax
  mov [402h],dx
  mov si,0
aff:mov dl, [400h+si]
  mov ah,02
  int 21h; affiche dl
  inc si
  cmp si,3
  jnz aff
  ret
   fact:
  pop bx
  pop ax
  cmp ax,0
  jz fin1
  mov cx,ax
etq: dec cx
  jz  fin2
  mul cx
  jmp etq
  fin1: mov ax,1
  fin2:push bx; sp
  ret
```

Exercice n°3 (05 points):

Donner un programme assembleur 8086 pour afficher la figure en face. Rappel: utiliser int 21h et ah=02 pour afficher un caractère. pour le retour à la ligne utiliser le code ASCII 13.



```
mov si,10
mov bx,0001h
debut:
mov cx,bx
ligne:
mov ah,02
mov dl,'*'
int 21h
loop ligne
mov dl,'\'
int 21h
mov dl,10; retour a la ligne
int 21h
mov dl,13; retour a la ligne
int 21h
inc bx
cmp bx,si
jnz debut
dec bx
debut2:
mov cx,bx
ligne2:
mov ah,02
mov dl,'*'
int 21h
loop ligne2
mov dl,'/'
int 21h
mov dl,10; retour a la ligne
int 21h
mov dl,13; retour a la ligne
int 21h
dec bx
jnz debut2
ret
```

Exercice n°4 (05 points)**:** soit le jeu d'instruction suivant d'un processeur possède 10 registres d'usage général de 32 bits (س0، س1، ...، س9), un registre compteur ordinal de 32bits (ع), un registre d'état de 8 bits (flag *zero*, flag *parité*, flag *signe*,..). La mémoire est organisée comme dans x86. Le bus de donnée a 32 fils. le bus d'adresse a 30 fils.

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.

انقل [100h]، س1 **انقل [104h]، س2** **انقل [108h]، س3** **انقل 1،س0** **اجمع س0، س1،س1** **اطرح س2،س0، س2** **اضرب س2،س1،س2** **اقسم س3،س2،س4** **انقل س4، [200h]** **انقل س2، [204h]**
La taille mémoire adressable=2^{30} octets=1 Gigat octets Le mot mémoire= 32 bits L'adresse de la dernière case= FFFF FFFF On a pas besoin de segmenter la mémoire, puisque le CO a 32 bits. On peut adresser plus de 1 Gigat octets (4 Gigat).

التمرين 01: هات برنامج مجمّع 8086 يطلب إدخال عدد صحيح n باستخدام لوحة المفاتيح، ومعرفة ما إذا كان زوجيًا أو فرديًا ، ثم يعرض النتيجة على الشاشة.

التمرين 02: أعطِ برنامج مجمّع 8086 لحساب مضروب عدد صحيح مخزّن في العنوان [300]. استخدم دالة تسمى *fact* الموجودة بالقرب من البرنامج الرئيسي. يجب أن يتم مرور العدد الصحيح بواسطة المكدس. يجب أن يتم إرجاع النتيجة عن طريق السجلات. تخزين النتيجة يكون ابتداء من العنوان 400. اعرض النتيجة على الشاشة.

التمرين 03: هات برنامج مجمّع 8086 لعرض الشكل المقابل. تذكير: استخدم int 21h و ah = 02 لعرض حرف. للعودة إلى السطر، استخدم رمز ASCII 13.

التمرين 04: لتكن مجموعة التعليمات التالية لمعالج يحتوي على 10 سجلات لأغراض عامة ذات سعة= 32 بت (س 0 ، س 1 ، ... ، س 9) ، سجل عداد البرنامج 32 بت (يسمى: ع) ، سجل حالة ذي 8 بت (علم الصفر ، علم الزوجية ، علم الإشارة ، ..). تم تنظيم الذاكرة كما في x86. يحتوي ناقل البيانات على 32 سلكا. يحتوي ناقل العنوان على 30 سلكًا.

1- هات برنامجًا للقيام بالعمليات التالية على 03 أرقام x و y و z المخزنة في [100] و [104] و [108] على التوالي. ثم احسب:

```
x:=x+1;
y:=y-1;
y:=y*x;
z:=z/y;
```

قم بحفظ نتيجة العملية الأخيرة في [200] و [204].

2- حدد لهذا المعالج: حجم الذاكرة التي يمكن عنونتها ، وكلمة الذاكرة ، وعنوان الخانة الأخيرة من الذاكرة. هل نحتاج إلى تقسيم الذاكرة إلى مناطق؟ علل.