1 : soit le programme suivant :

- récrire le programme mais en commentant chaque ligne
- compiler et linker puis exécuter pas à pas le programme sur « **td** » de TASM
- Quelles sont les valeurs initiales et finales prises par les registres
 - AX, CX, IP
- quels sont les flags qui sont modifiés par cette exécution ?
- Quelle est le rôle de « adr »

changer la valeur de CX par 8, quelle la valeur finale de IP, AX ?

```

pile segment stack
db 0100h dup (?)
pile ends
data segment
data ends
assume ds : data, cs : code, ss : pile
code segment
debut:    mov ax, 12h
          mov cx, 10
          adr:
          add ax, ax
          loop adr
          mov ah, 4ch
          int 21H
code ends
end debut
    
```

2. Exécuter ce code sous Td de tasm

- Quelle sont les valeurs des flags ?
- Expliquer ces valeurs ?
- Que vaut 8000h en binaire et en décimal ?
- de même pour 9000h ?
- Donner le résultat d'exécution en binaire puis en décimal

```

pile segment stack
db 010h dup (?)
pile ends
assume ds : code, cs : code, ss : pile
code segment
debut: mov ax, 8000h
          add ax, 9000h
          mov ah, 4CH
          int 21H
code ends
end debut
    
```

3. Exécuter ce code sous td de tasm :

- Observer l'exécution et quelle est la valeur finale de CX et flags ?
- donner à BX la valeur 2
- Re-observer l'exécution. flags= ? et CX= ?
- Quelle est le rôle 'jz' et 'jump' ?
- Donner l'algorithme correspondant
-

```

pile segment stack
db 0100h dup (?)
pile ends
assume ds : code, cs : code, ss : pile
code segment
debut: mov ax,2
          mov bx,1
          sub ax,bx
          jz egal
          sub cx,0
          jmp fin
egal: mov cx,1
fin: mov ah, 4CH
int 21H
code ends
end debut
    
```

N.B. 1. A remettre un rapport en papier et Attention aux rapports identiques !!! ☹

2. Travailler en monôme ou binôme et pas plus

Bon travail