

ATTENTION

- TOUT document NON autorisé
- TOUT échange (documents ou autres biens) INTERDIT (effaceur, etc..)
- TOUS équipements électroniques ETTEINTS (phones, calculatrices . .)

Exercice 1 (5 Pts)(Temps_référence : 20 mn)**NB : il n'est pas nécessaire de reproduire les tables de Karnaugh !**A/- Trouver la forme simplifiée de la fonction (F_1) définie par la table de Karnaugh suivante :

\ a b c	0 0 0	0 0 1	0 1 1	0 1 0	1 1 0	1 1 1	1 0 1	1 0 0
d e								
0 0	1	1	1	1	1	1	1	1
0 1	1	1	1	1	1	1	1	1
1 1	1	1	1	1	1	1	1	1
1 0	1	1	1	1	1	1	1	1

B/- Trouver la forme simplifiée de la fonction (F_2) définie par la table de Karnaugh suivante :

\ a b c	0 0 0	0 0 1	0 1 1	0 1 0	1 1 0	1 1 1	1 0 1	1 0 0
d e								
0 0	0	0	0	0	0	0	0	0
0 1	0	0	0	0	0	0	0	0
1 1	0	0	0	0	0	0	0	0
1 0	0	0	0	0	0	0	0	0

C/- Trouver la forme simplifiée de la fonction (F_3) définie par la table de Karnaugh suivante :

\ a b c	0 0 0	0 0 1	0 1 1	0 1 0	1 1 0	1 1 1	1 0 1	1 0 0
d e								
0 0	0	0	1	0	1	0	0	0
0 1	0	0	1	0	1	0	0	0
1 1	0	0	0	0	0	0	0	0
1 0	0	0	0	0	0	0	0	0

D/- Pour la forme simplifiée (F_3), donner l'expression en NAND à 2 entrées exclusivement.**Exercice 2 (10 Pts)**(Temps_référence : 20 mn)

On considère le programme ci-contre :

1)- Si on suppose ($n=10$) : indiquer la fonction de ce programme c-à-d : que fait ce programme (en 2 phrases) ?2)- Indiquer la nouvelle valeur de (n) si on souhaite atteindre les 2 objectifs suivants :

A/- on transfère 256 données

B/- les données transférées sont superposées en RAM (adresses successives)

☞ Réécrire le programme avec les modifications demandées.

```

START:  MOV CX, 80H
        PUSH BX
        MOV AX, [BX]
        MOV [DI], AX
        ADD DI, 2
        ADD BX, n
        LOOP START

```


Exercice 3 (5 Pts)(Temps_référence : 20 mn)

Proposer un programme assembleur '80286' qui effectue les opérations suivantes :

* charge successivement 1024 données, de 2 octets chacune, à partir d'une adresse (SI) quelconque ;

* identifie le 15^{ème} bit de chaque donnée chargée :

* si ce 15^{ème} bit est nul, la donnée est stockée dans une zone RAM pointée par DI quelconque ;

* si ce 15^{ème} bit est non nul, la donnée est stockée en pile.

Attention : On suppose pour cette question que les octets des données sont inversés en RAM, c-à-d classées en zone (SI) comme suit :

L'@ (SI+0) contient le MSB (Octet de poids Fort) de la 1^{ère} donnée Data_0 ;

L'@ (SI+1) contient le LSB (Octet de poids faible) de la 1^{ère} donnée Data_0 ;

L'@ (SI+2) contient le MSB de la 2^{nde} donnée Data_1 ;

L'@ (SI+3) contient le LSB de la 2^{nde} donnée Data_1 ;

.. etc ...

(il faudra donc réordonner les octets de chaque donnée)

Bon Courage