

EXAMEN DE RATRAPAGE

- 1- Qu'est ce qu'une étiquette ? Qu'elles sont les règles d'utilisation d'une étiquette ?  
(3 points)
- 2- Dans la machine de Babbage, soit  $y = f(x)$  (4 points)  
Quelles sont les étapes à suivre et les données nécessaires pour calculer  $D_i$  de la ligne  $n$  et sachant que la ligne 1 correspond à  $x = 1$  etc...  
Pour :
  - a-  $i > n$
  - b-  $i < n$
- 3- Nous avons 4 nombres entiers en mémoire U, V, Y, T. si la somme de 2 de ces 4 nombres est égale à la somme des 2 autres, on affectera cette somme à chacune des 4 variables, sinon on leur affectera 0. On vous demande d'écrire le programme assembleur qui réalisera ce travail (5 points)
- 4- Organigramme (6 points) :  
Soit un tableau d'entier A de 4 lignes et 4 colonnes et un tableau d'entier B à une dimension dont le nombre d'éléments est égale à la moitié du nombre d'éléments du tableau A. On remplit le tableau A comme suit : si la somme des indices d'un élément du tableau est paire, alors on met cette somme d'indice dans l'élément. Sinon on met son indice de ligne. Ensuite on remplit les éléments du tableau B par les éléments du tableau A ligne par ligne (les premières lignes). Donnez l'organigramme qui remplit les tableaux A et B.
- 5- Quel est l'intérêt de l'utilisation des sous programmes ? Indiquez les différents types de sous programmes et en quoi ils diffèrent (1 point)  
Expliquez par un exemple le principe d'utilisation d'un sous programme (1 point)

Corrigé type

## Examen de rattrapage

- 1- Une étiquette représente l'adresse mémoire de l'instruction. **(0.5 pt)**
  - Une étiquette doit apparaître une seule fois à gauche de l'instruction **(1 pt)**
  - Elle peut être utilisée plus d'une fois en tant qu'opérande d'une instruction de branchement **(1 pt)**
  - Une étiquette ne doit pas apparaître dans la même instruction de branchement en tant qu'étiquette et opérande sinon on a une boucle infinie **(0.5 pt)**
- 2-
  - a- Pour  $i > n$ , il n'y a pas de valeur de  $D_i$  (ex : si  $i = 3$  et  $n = 2$  on a  $D_2$  mais pas de  $D_3$ ) **(1 pt)**
  - b- Pour  $i < n$  on doit : **(3 pts)**
    - Commencer à la ligne  $n-i$  ( $x = n-i$ ) afin de calculer  $y$
    - Ensuite on passe à la ligne  $n-i+1$  afin de calculer  $D_1$   
(Sachant que  $D_1$  de la ligne  $(n-i+1) = y$  de ligne  $n-i+1 - y$  de ligne  $(n-i)$ )
    - Ensuite on passe à la ligne  $n-i+2$  afin de calculer  $D_2$   
(Sachant que  $D_2$  de la ligne  $(n-i+2) = D_1$  de ligne  $(n-i+2) - D_1$  de ligne  $(n-i+1)$ )
    - Etc...
    - Jusqu'à la ligne  $n-i+i=n$  afin de calculer  $D_i$  où :  
 $D_i$  de ligne  $(n) = D_{i-1}$  de ligne  $(n) - D_{i-1}$  de ligne  $(n-1)$
- 5- \*Si dans un programme il y a une suite séquentielle d'instructions qui doit être exécutée en différents endroits, afin d'éviter de réécrire cette suite, alors on l'utilise comme sous programme **(0.5 pt)**
  - \*Fonction qui retourne un résultat et procédure qui ne retourne pas de résultat **(0.5 pt)**
  - \*On définit le sous programme (type, nom, arguments et corps) au début, et on fait appel au point d'exécution (nom, paramètres) **(1 pt)**

3/ (5 pts)

```
LDA      U
ADD      V
STA      Z
LDA      Y
ADD      T
SUB      Z
BRZ      ET1
LDA      U
ADD      Y
STA      Z
LDA      V
ADD      T
SUB      Z
BRZ      ET1
LDA      U
ADD      T
STA      Z
LDA      V
ADD      Y
SUB      Z
BRZ      ET1
LDA      #0
BRI      ET2
ET1 LDA   Z
ET2 STA   U
      STA  V
      STA  Y
      STA  T
      END
```

- structure du programme (1 pt)
- branchements (2 pts)
- autres instructions (1 pt)

Exercice 4 :  
( 6 pts )



