

UTBM

Median AG43

Automne 2009

Réduction d'un Buffer

On considère un buffer d'octets dont la longueur ne peut excéder 255 Octets. Le format du buffer est le suivant :

```
[longueur n][octet 1][octet 2]...[octet n]
```

n est inférieur à 256.

Ce buffer est utilisé pour manipuler des nombres entiers codés en base 100.

Il arrive souvent, qu'après une opération comme une soustraction, la valeur contenue dans le buffer soit de plusieurs ordres de grandeur inférieure à sa valeur initiale. Dans ce cas, les premiers octets (hormis l'octet 0 qui donne la longueur utile du buffer) contiennent la valeur 0. On souhaite écrire une procédure appelée `reduction(buffer)` qui supprime les premiers octets nuls, tout en conservant l'intégrité du format utilisé.

Comptage de mots

Certains Logiciels de traitement de texte sont pourvus d'une fonction de comptage de mots, qui est utile notamment pour évaluer la longueur d'un résumé.

On vous propose d'écrire cette fonction avec le cahier des charges (volontairement simplifié) suivant :

- Le texte est placé dans une mémoire (tableau de caractères). La fin du texte est repérée par le caractère NUL (valeur 0). Ce tableau sera appelé "texte".
- Les mots sont séparés par un caractère espace que nous appellerons ESP, par un caractère retour chariot, désigné par CR, ou par un caractère de tabulation désigné par TAB. Attention, le texte peut commencer par un ou plusieurs de ces caractères.
- Les autres caractères comme les apostrophes seront traités comme des caractères normaux ("l'horizon" sera compté comme un seul mot).

La fonction sera appelée de la manière suivante :

```
nombre_de_mots = compte_mots(texte)
```

ce qui signifie qu'elle retournera un entier qui sera le nombre de mots que contient le tableau passé en paramètre.

Conversion binaire en hexadécimal codé ASCII

Il arrive souvent d'afficher le contenu d'un octet (qui contient un nombre binaire de 8 bits) sous la forme d'un groupe de deux caractères hexadécimaux. Chaque caractère hexadécimal est le représentant d'un groupe de 4 bits appelé quartet qui est la moitié supérieure ou inférieure de l'octet. On rappelle que les caractères hexadécimaux sont : 0, 1, ..,9, A, .., F, et que la valeur décimale représentée par un quartet est comprise entre 0 et 15.

Ecrire une fonction appelée `Hex(a)`, admettant comme argument un octet (entier sur un octet, non signé), et retournant un caractère ASCII qui sera la représentation hexadécimale codée ASCII de l'argument. L'argument possède le quartet de poids fort toujours nul, ce qui signifie que la valeur de `a` est comprise entre 0 et 15, et que l'on n'aura pas à effectuer de vérification.

Pour mémoire, le code ASCII du caractère '0' (zéro) est 48, celui du caractère 'A' est 65.

Feuille à détacher et à remplir**Nom :****Prénom :****Signature :****Questions FORTRAN**

```
program test1
implicit none
real S ;integer k
do k=1,3
    S=S+1/k
enddo
write(*,*) S
end program test1
```

Question: que vaut S lors de l'affichage?

Réponse :

```
program test2
integer*1 k ;character car
equivalence(k,car)
car='b'
k=k/2
write(*,*) car
end program test2
```

Question: que vaut car lors de l'affichage?

Réponse :

```
program test3
integer i,A(2,2) /1,2,3,4/
do i=1,2
    write(*,'(2I2,$)') (A(i,j),j=1,2)
enddo
end program test3
```

Question: Qu'est qui s'affiche à l'écran sachant que le remplissage de A s'effectue colonne par colonne? (le second indice de la matrice A étant considéré comme l'indice de colonne)

Réponse :

```
program test4
real*4 A(3,2) /1,2,3,4,5,6/
open(12,file='test.txt')
do i=1,3
    write(12,'(2F16.7)') (A(i,j),j=1,2)
enddo
close(12)
end program test4
```

Question: Quelle est la taille du fichier 'test.txt' en octets sachant que chaque appel à « write » place 2 caractères supplémentaires en fin de ligne dans ce cas?

Réponse :

```
program test5
real*4 A(3,2) /1,2,3,4,5,6/
open(12,file='test.bin',form='binary')
do i=1,3
    write(12) (A(i,j),j=1,2)
enddo
close(12)
end program test5
```

Question: Quelle est la taille du fichier 'test.bin' en octets?

Réponse :