Université Libanaise
Faculté des Sciences 1

INFO216 F

Programmation Impérative II

Session2 15,16
3 Septembre 2016
Durée: 2 heures

Les trois questions suivantes sont indépendantes et toutes utilisent le même type de données "nœud" suivant:

```
typedef struct noeud {char c; noeud * suivant;} noeud;
```

Exercice I [40 min - 20 points]

Dans une syntaxe d'expression régulière simplifiée, quand une lettre est suivie par un astérisque (*), cela signifie que cette lettre est répétée zéro, une ou plusieurs fois de suite. Et quand elle est suivie d'un chiffre, cela signifie que la lettre est répétée exactement ce nombre de fois.

Ecrire une fonction "match" qui, étant donné une liste simplement chaînée L de noeuds et une expression régulière RE représentée par une chaîne de caractères, détermine si oui ou non la chaîne décrite par L correspond à RE. Nous supposons que l'alphabet est limitée aux lettres de 'a' à 'z', et que chaque lettre dans RE est suivi par un '*' ou un chiffre.

```
Exemples:
RE = "a2b*" → quelques échantillons: "aa", "aab", "aabb", "aabbb", etc.
RE = "b*a*c1" → quelques échantillons: "c", "bc", "ac", "aac", "abc", etc.
```

Exercice II [40 min - 25 points]

Un fichier texte "lettres" contient un texte en anglais.

1. Définir un nouveau type de structure "noeud2", pour une liste simplement chaînée, capable de tenir un caractère et un compteur en tant que champs de données.
2. Ecrire la fonction "stat" qui lit le fichier "lettres", construit et retourne une liste simplement chaînée de "noeud2" contenant les statistiques en nombre d'occurrences pour chaque lettre (a→z). Un exemple est donné ci-après.

```
Exemple:
```

lettres

```
Eva works at a
gallery cafe
in London. She
sees many
paintings.
```

Liste:

```
a,7→ c,1→ d,1→ e,6→ f,1→ …→ NULL
```

Exercice III [40 min - 25 points]

Étant donné une liste simplement chaînée de noeuds et un nombre k, écrire une fonction "renverser" qui renverse en alternance (une fois sur 2) k noeuds de la liste. Examinez attentivement l'exemple ci-dessous.

```
Exemple:
Liste : a b c d e f g h i j k l m n o p q r null, k = 4
Sortie : d c b a e f g h l k j i m n o p r q null
```