Lebanese University
Faculty of Sciences 1

INFO216

Imperative Programming II

Session 2 08.09
(Friday 11 Sep. 09)
Duration: 2 hours

1. Five full marks will be reserved for the good presentation of the answer sheet.

The following three questions are independent. Though, they all use the same data type "list":

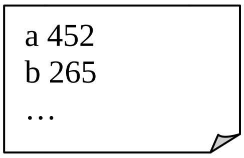
```
typedef struct list{ char c; list *next;} list;
```

2. Write a recursive function "reverse" that reverses a single linked list recursively. This function takes a single linked list as argument and reverses it and returns back the head node to the reversed list.

3. In a simplified regular expression syntax, a * following a letter means that this letter could appear zero, one or many times consecutively, and a digit following a letter means that the letter should appear exactly this number of times. Given a linked list L, and a regular expression RE represented by a string, write a function "match" that determines whether the string in the list L is a member of the language defined by RE.

- Examples:
  1. RE="a2b*" → sample members of this language: "aa", "aab", "aabb", "aabbb", etc.
  2. RE="b*a*c1" → sample members of this language: "aac", "abc", "c", "ac", "aaac", etc.
- Note: we suppose that
  1. there are only the * operator and the digit operator in regular expressions,
  2. the alphabet is restricted from letters a to z,
  3. every letter will be followed by either * or a digit

4. A binary file "letters" contains the letters of an English article. Each letter put inside a "list" element was written using the function fwrite. Write the function count that reads the file "letters", constructs a single linked list of a new type adding the field "occurrences", then creates and saves the text file "stat" holding the statistics for each letter (a → z). Each line in this file will contain a letter and its number of occurrences in the article. An example is given below.



```
a 452
b 265
…
```

The End