

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

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

### Exercise I [40 min – 20 points]

In a simplified regular expression syntax, when a letter is followed by an asterix (\*), this means that this letter is repeated zero, one or many times consecutively. And when it is followed by a digit, it means that the letter is repeated exactly this number of times.

Write a function "match" that, given a linked list L of nodes, and a regular expression RE represented by a string, determines whether or not the letters in L matches RE. We suppose that the alphabet is restricted to the letters 'a' to 'z' and that every letter in RE is followed by either \* or a digit.

Examples:

RE="a2b\*" → sample matches : "aa", "aab", "aabb", "aabbb", etc.

RE="b\*a\*c1" → sample matches : "c", "bc", "ac", "aac", "abc", etc.

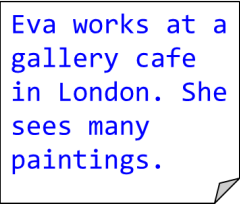
### Exercise II [40 min – 25 points]

A text "letters" contains an English text.

1. Define a new structure type "node2", for a singly linked list, to hold a character and a count as data fields.
2. Write the function "stat" which reads the file "letters", constructs and returns a singly linked list of "node2" holding the statistics for each letter (a → z). An example is given below.

Example:

letters



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

List:

a,7 → c,1 → d,1 → e,6 → f,1 → ... → NULL

### Exercise III [40 min – 25 points]

Given a linked list of nodes and a number k, write a function "reverse" which reverses every alternate k nodes of the list. Examine thourouly the example below.

Example:

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

Output : d c b a e f g h l k j i m n o p r q null

**The End**