Exercise III

Implementation of text operations

The language MINI has to be extended with a number of operators for TEXTs, i.e. they have to be added to the standard prelude and to the runtime system:

```
TEXT OP + (TEXT CONST a, TEXT CONST b)        (concatenation)
     OP CAT (TEXT VAR a, TEXT CONST b)        (concatenation)
TEXT OP * (INT CONST a, TEXT CONST b)         (replication)
INT  OP LEN (TEXT CONST a)                    (length)
INT  PROC length (TEXT CONST a)               (length)
TEXT OP SUB (TEXT CONST a, INT idx)           (subscription)
BOOL OP =  (TEXT CONST a, TEXT CONST b)       (comparison)
BOOL OP <> (TEXT CONST a, TEXT CONST b)       (comparison)
BOOL OP <  (TEXT CONST a, TEXT CONST b)       (comparison)
BOOL OP <= (TEXT CONST a, TEXT CONST b)       (comparison)
BOOL OP >  (TEXT CONST a, TEXT CONST b)       (comparison)
BOOL OP >= (TEXT CONST a, TEXT CONST b)       (comparison)
```

Examples:

```
put (nr * " " + text (ivar) + ": " + text (result) + "\n");
b CAT bw * " new " + text (bvar);
LEN input < 40
```

Mark that these additions do not require a change to the syntax rules or semantics of the language. They merely involve additions to the runtime system.

In ELAN, TEXTs start with index 1 (not 0). This means that to ensure compatibility, the SUB operator has to be implemented in such a way that

```
"abcdef" SUB 2 = "b"
```

Furthermore the SUB operator yields the empty text when the index is out of range.

A small set of test programs can be obtained by unpacking the file `ftp.cs.kun.nl:/pub/cdl3/ex3.tgz` in your MINI working directory.