

Programmazione ad Oggetti

Homework 5 (implementazione della classe stringa)

Implementare i metodi della classe Stringa allegata. Alla specifica della classe vanno aggiunti gli opportuni qualificatori di accesso al fine di implementare una corretta politica di information hiding.

Si aggiunga inoltre il qualificatore const nei seguenti casi:

- funzioni membro che non modificano lo stato dell'oggetto
- parametri che non vengono modificati dalle funzioni membro alle quali vengono passati

Specifiche della classe

- La classe gestisce in maniera dinamica ed automatica l'array di caratteri, aumentando e diminuendo in maniera opportuna le sue dimensioni, allo scopo di contenere esattamente i caratteri richiesti. Questo aspetto dovrà essere tenuto in conto nell'implementazione delle funzioni `insert()` e `operator+=()`.

- L'array di caratteri è gestito SENZA usare il carattere terminatore `'\0'`, ma bensì usando come informazione la variabile `num`.

- Le funzioni di accesso e modifica controllano la correttezza dei parametri passati.

ES.

data la stringa

```
Stringa s("Pippo");
```

la chiamata

```
s.get(7);
```

dovrà generare un opportuno messaggio di errore e terminare il programma.

- Per il **confronto tra due stringhe** si usi il seguente algoritmo:
 - `n=1` (0 in C++)
 - si confrontano i simboli nella posizione n-esima della stringa:
 - se i simboli sono uguali, si passa alla posizione successiva della stringa ($n \rightarrow n+1$)
 - se una delle due stringhe non possiede l'elemento n-esimo, allora è minore dell'altra e l'algoritmo termina
 - se entrambe le stringhe non possiedono l'elemento n-esimo, allora sono uguali e l'algoritmo termina
 - se questi sono diversi, il loro ordine è l'ordine delle stringhe
- si definisca inoltre una lunghezza massima possibile per la stringa. Questa grandezza dovrà essere memorizzata come variabile `const static`. Nel caso in cui lo spazio necessario ecceda questo valore, è necessario stampare un opportuno messaggio di errore.

Suggerimenti

- per l'implementazione dell'algoritmo di confronto del punto precedente si implementi la seguente funzione:

```
int compare (const char* s1, const char* s2)
```

che restituisce:

0 se le due stringhe sono uguali

+1 se `s1 > s2`

-1 se `s1 < s2`

- si implementi una funzione di copia:

```
int copy(char *dst, const char *src, const unsigned int len)
```

che copia la stringa `src` nella stringa `dst`. Il comportamento della funzione è il seguente:

se `len == 0` copia dalla sorgente al destinazione finché non trova il terminatore nella sorgente

se `len > 0` copia dalla sorgente al destinazione `len` caratteri

- si implementi infine la funzione:

unsigned int length(**const char** *str)

che restituisce il numero di caratteri contenuti nella stringa passata come parametro (contiene il terminatore).