

# Programmazione I

## Prova di programmazione – 9 giugno 2017 – 2 ore

Date due parole (stringhe non contenenti spazi) **s1** ed **s2**, di lunghezza **n1** ed **n2**, la parola **s2** è un prefisso/suffisso per la parola **s1** se **n2**  $\geq$  **n1**, e la sottoparola di **s1** data dai suoi primi/ultimi **n2** caratteri è uguale ad **s2**. Partendo da questa definizione e dal frammento di codice fornito, realizzare un programma per la memorizzazione di un insieme di parole con prefissi comuni, ossia di parole tali che ciascuna parola o è un prefisso per le altre parole di lunghezza maggiore della sua, o è la parola più lunga (maggiori dettagli nella descrizione delle funzionalità). Il numero massimo di parole che possono essere inserite è definito a tempo di scrittura del programma. Memorizzare ciascuna parola in modo tale da **occupare meno spazio possibile in memoria**, sfruttando anche il fatto che tutte le parole di lunghezza minore, già presenti nell'insieme, sono un suo prefisso.

All'avvio del programma, l'insieme è vuoto. Se può essere utile, la sintassi per definire un array di puntatori a carattere (che può poi essere interpretato come un array di stringhe dinamiche) è **char \*<nome\_array>[NUM\_ELEM\_ARRAY];**

mentre, per allocare l'array dinamico puntato dall'i-esimo elemento di questo array,

```
<nome_array>[i] = new char[<lunghezza>];
```

infine, per accedere all'elemento j-esimo dell'array i-esimo, la sintassi è

```
<nome_array>[i][j]
```

Il programma deve fornire le seguenti funzionalità.

1. **aggiungi\_parola(suff)** Aggiunge una nuova parola **s**, uguale alla parola più lunga correntemente memorizzata, più il suffisso **suff**. Ad esempio, se l'insieme è vuoto e **suff** è la parola "btc", nell'insieme viene inserita la parola "btc". Se invece l'insieme già contiene, ad esempio, le parole "gh" e "ghsty", allora verrebbe aggiunta la parola "ghstybtc".
2. **stampa\_parole** Stampa le parole nell'insieme, in ordine di lunghezza crescente. Ad esempio, per l'insieme al secondo esempio nel precedente punto, stamperebbe  
gh  
ghsty  
ghstybtc
3. **salva\_stato** Salva il contenuto dell'insieme in un file di testo dal nome definito a tempo di scrittura del programma.
4. **carica\_stato** Carica il contenuto della sequenza dal file. Il contenuto precedente è perso.
5. **controlla\_se\_prefisso(p)** Controlla se, tra le parole correntemente memorizzate nell'insieme, la parola **p** risulta essere un prefisso per almeno una parola di lunghezza strettamente maggiore di quella di **p**. In caso di successo, ritorna l'indice della più piccola parola nell'insieme tale che **p** è un suo prefisso, assumendo che l'indice zero sia relativo alla parola più corta, e che gli indici delle parole crescano al crescere della lunghezza delle parole stesse. Ad esempio, per l'insieme di esempio al punto 2, se venisse passata la parola "ghst", ritornerebbe 1.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'*overflow* e l'inserimento di dati in formato errato e di parole troppo lunghe da *stdin*.

---

### REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
  - a) il programma è perfettamente funzionante in ogni sua parte
  - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati