

### Esercizio 1.

Si scriva un programma C++ opportunamente modularizzato in funzioni che, letta da input una frase, stampi una sottostringa della parola più lunga, determinata secondo il seguente criterio: dalla parola più lunga della frase vengono eliminate, nell'ordine in cui compaiono, quelle lettere che corrispondono alle prime lettere delle altre parole della frase.

Esempio: "non mi piace guidare in montagna o in collina".

- La parola più lunga della frase è "montagna".
- Le prime lettere delle altre parole sono: n(on), m(i), p(iace), g(uidare), i(n), o, i(n), c(ollina).
- Da "montagna", si eliminano nell'ordine la "n", la "m", la "g" e la "o": montagna → motagna → otagna → otana → tana.
- Si stamperà la parola "tana".

### Esercizio 2.

Si scriva un programma C++, opportunamente modularizzato in funzioni, che implementi il semplice (si legga stupido) correttore ortografico descritto di seguito. Viene fornito in input un dizionario (elenco di parole), l'utente digita una frase, terminata dal carattere '.' e il correttore, per ogni parola inserita P, verifica che P sia presente nel dizionario;

- se P è presente, la parola viene stampata in output così com'è;
- se invece P non è presente, il correttore cerca nel dizionario una possibile correzione (ad esempio la prima in elenco), ossia una parola che differisca da P SOLO per una lettera e stampa la correzione al posto di P.
  - Se nessuna correzione viene trovata nel dizionario, P viene stampata così com'è.

Esempio: L'utente digita la frase "Quel ramo del lagi di Coma." Il correttore trova nel dizionario le parole Quel, ramo, del, di, ma non trova le parole lagi e Coma. Cerca quindi una possibile correzione per lagi e trova magi, ma non trova alcuna correzione per Coma. L'output dello stupido correttore sarà perciò la frase: "Quel ramo del magi di Coma."