

Programmazione I
Prova di programmazione – 21 Luglio 2021 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma per la gestione di un *crucipuzzle*. Il crucipuzzle è quadrato, ossia ha numero di colonne N uguale al numero di righe. Il valore di N è deciso e può cambiare a tempo di esecuzione del programma. I caratteri nel puzzle possono essere solo lettere maiuscole. All'avvio del programma, il puzzle ha dimensioni 0×0 . Il programma fornisce le seguenti funzionalità.

1. **inizializza_puzzle(N)** Inizializza il puzzle a dimensioni $N \times N$ e ne legge il contenuto da *stdin*. In particolare, la funzionalità legge tale contenuto come un'unica sequenza di caratteri, in ordine di riga. L'eventuale precedente contenuto del puzzle è perso. Qualsiasi carattere immesso dall'utente che non sia una lettera maiuscola è convertito in una **A**, senza alcuna segnalazione di errore.
2. **stampa_puzzle** Stampa il nome il contenuto del puzzle, con i caratteri separati da spazi. Ad esempio, per un puzzle 4×4 potrebbe stampare:
A G O E
M O I W
I L M L
C A S A
3. **[2] salva_puzzle** Salva il puzzle in un file di testo.
4. **[3] carica_puzzle** Carica il puzzle dal file di testo. Il precedente puzzle è perso.
5. **[2, 3] cerca_parola_verticale(p)** Cerca la parola **p** in verticale, verso il basso o verso l'alto nel puzzle. Stampa PRESENTE se la trova, ASSENTE se non la trova o se, come caso particolare, la parola **p** ha lunghezza 0. Ad esempio, dato il contenuto del puzzle riportato nell'esempio al punto 2, stampa PRESENTE per le parole MIO, IO, GOL A e CIMA; stampa invece ASSENTE per le parole CASA, AGO o LAMA. Si ottiene il voto massimo se l'implementazione di questa funzionalità non controlla mai inutilmente sequenze di caratteri che sarebbero troppo brevi per poter coincidere con la parola **p**.
6. **[4, 7] cerca_parola_globale(p)** Controlla se nel puzzle ci sono i caratteri della parola **p**, ovunque essi siano. Stampa PRESENTE se li trova, ASSENTE altrimenti. Stampa ASSENTE anche nel caso in cui la parola **p** abbia lunghezza 0. Si ottiene il punteggio massimo se si scandisce il puzzle una sola volta.

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 da *stdin*.

Per compatibilità con il tester, per le funzionalità 5 e 6 assicurarsi di stampare un *newline* prima della stampa dell'esito della ricerca, altrimenti eventuali caratteri precedenti, non seguiti da *newline*, andrebbero a fondersi con tale stampa. Inoltre assumere che la parola **p** possa essere lunga almeno 10 caratteri.

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo 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
 - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale