

Fondamenti di Programmazione

A.A. 2020/2021

Appello del 21 Luglio 2021

Il codice sorgente che contiene lo svolgimento della prova d'esame deve essere memorizzato in un singolo file (es: *cognome.c*). La consegna di tale codice sorgente avviene tramite la pagina del corso sul portale USienaIntegra: <https://elearning.unisi.it/course/view.php?id=4128>. E' necessario e sufficiente consegnare soltanto il file sorgente.

PROVA D'ESAME

Scrivere il programma che calcola alcune statistiche relative ad un insieme di stringhe memorizzate sul file `input.txt`. Le statistiche sono calcolate confrontando ogni stringa del file rispetto ad un pattern noto, che assumiamo sia "ABCD".

La prima riga del file memorizza un intero che indica il numero di righe successive, mentre ogni riga successiva alla prima memorizza una stringa di lettere maiuscole.

Un esempio di file è il seguente:

```
3
ADDBACDBABDDCADD CBAAA
DBCABABLCBABCD CDC
AFBBBDKCLMNOABABCD CDOBST
```

Per ciascuna stringa, il programma calcola una matrice binaria `matrix` che ha lo stesso numero di righe della lunghezza del pattern noto e lo stesso numero di colonne della lunghezza della stringa considerata. L'elemento della matrice di posizione i,j è 1 se l' i -esimo elemento del pattern noto è uguale al j -esimo elemento della stringa, altrimenti è 0.

Considerando la prima stringa del file dell'esempio precedente, otteniamo:

```
1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1
0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
0 1 1 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0
```

Utilizzando tale matrice il programma calcola la frequenza (assoluta) di ciascun carattere del pattern noto nella stringa e se il pattern intero occorre almeno una volta nella stringa (anche se distribuito in posizioni non contigue), e stampa a video il riassunto delle statistiche calcolate. Considerando sempre la prima stringa del file dell'esempio, il formato di stampa è il seguente:

```
Statistiche per la sequenza: ADDBACDBABDDCADD CBAAA
la frequenza di A e': 7
la frequenza di B e': 4
la frequenza di C e': 3
la frequenza di D e': 7
```

Si richiede che i dati relativi ad una stringa vengano memorizzati in una struttura (`struct`) i cui elementi chiamati `testo`, `freq`, `occur` (dei tipi opportuni) permettano di memorizzare la stringa, la frequenza di ciascuna lettera del pattern noto e un valore binario che indichi la presenza o meno di una occorrenza dell'intero pattern nella stringa. Si definisca un nuovo "tipo" associato alla struttura (nome del nuovo tipo: `statistica`).

Si assume che: la lunghezza massima (TLEN) di una stringa è 50, la lunghezza (PLEN) del pattern dato è 4, e il numero massimo di stringhe memorizzate nel file è 100.
Non è permesso l'utilizzo di variabili globali.

Esercizio 1 [punti 6]

`main()`

Invoca le funzioni degli esercizi seguenti (in ordine).

Tutte le variabili passate come parametri alle funzioni devono essere dichiarate in `main`.

Il file `input.txt` su cui sono memorizzate le stringhe da analizzare viene aperto e poi chiuso in `main` dopo essere stato utilizzato. Se il file non esiste, il programma termina con un messaggio di errore di apertura del file, altrimenti, viene letta la prima riga del file per controllare che il numero sia valido (ovvero sia un numero positivo e minore di 100); nel caso in cui il numero sia maggiore di 100, soltanto le prime 100 stringhe del file saranno considerate dal programma.

Vengono ripetutamente eseguite le seguenti azioni, considerando una stringa del file alla volta:

- Viene invocata la funzione dell'Esercizio 2 che legge una stringa e calcola la matrice `matrix`. Se la funzione dell'Esercizio 2 ritorna 0 significa che c'è stato un errore di lettura e quindi il file non viene ulteriormente processato e il programma termina.
- Altrimenti, la stringa letta viene memorizzata nella struttura di tipo `statistica` e poi viene chiamata la funzione dell'Esercizio 3 che calcola una serie di statistiche aggiornando tale struttura.
- Infine, viene invocata la funzione dell'Esercizio 4 per la visualizzazione dei risultati.

Esercizio 2 [punti 10]

`leggi_seq(...)`

Legge una stringa dal file e costruisce la matrice stringa-pattern.

Legge una stringa dal file `input.txt`, e la memorizza in un array `testo` (parametro in uscita). Prende inoltre come parametri il puntatore al file e il pattern `pattern` (in ingresso), lunghezza della stringa letta e la matrice `matrix` stringa-pattern (parametri in uscita). La matrice ha numero di righe uguale a `PLEN` e numero di colonne uguale alla stringa di dimensione massima possibile, anche se il numero di colonne effettivamente utilizzate dipende dalla stringa letta. L'elemento della matrice di posizione i,j è uguale a 1 se l' i -esimo carattere del pattern è uguale al j -esimo carattere della stringa, altrimenti è uguale 0. La funzione restituisce 1 se la lettura è avvenuta con successo, altrimenti restituisce 0.

Esercizio 3 [punti 10]

`calcola_stat(...)`

Calcola le statistiche stringa-pattern (per una data stringa).

La funzione riceve come parametri in ingresso la matrice stringa-pattern e la lunghezza della stringa, mentre come parametri in uscita riceve una struttura `statistica` relativa alla stringa considerata. Calcola per ciascun carattere del pattern la sua frequenza nella stringa usando la matrice in ingresso: ogni riga contiene un 1 in corrispondenza delle posizioni relative allo stesso carattere del pattern nella stringa.

Per verificare se il pattern occorre per intero almeno una volta nella stringa (anche coprendo posizioni non continue), si scorre la prima riga della matrice alla ricerca di un elemento uguale a 1. Supponiamo che venga trovato in posizione i,j , con $i = 0$. Si scandiscono adesso gli elementi della seconda riga ($i=1$), partendo dalla colonna $j+1$ (attenzione, non si riparte dalla prima colonna!), di nuovo alla ricerca di un elemento uguale a 1. Si passa alla riga successiva e si ripete la procedura, fino ad esaurire tutte le righe. Il pattern occorre nella stringa se è stato trovato un 1 in tutte le righe della matrice.

Esercizio 4 [punti 6]

`stampa_stat(...)`

Visualizza le statistiche relative ad una stringa.

Riceve come parametri in ingresso una struttura di tipo `statistica` e il pattern noto. Visualizza le statistiche nel formato mostrato nell'esempio.