



**Politecnico  
di Torino**

# Esame Tecniche di Programmazione

*Appello 03/07/2021*

Andrea Grillo, S282802

Nella presente relazione verranno discussi gli approcci e le strategie adottate per la risoluzione degli esercizi di programmazione, con particolare riguardo agli errori commessi e alle relative correzioni.

Nota. Durante la scrittura nell'editor fornito, è risultato impossibile inserire le parentesi chiuse graffe '}' e quadre ']', in quanto alle relative combinazioni di tasti era associata la funzione di zoom della finestra. Per questo motivo, in sede d'esame si è fatto uso della parentesi chiusa tonda ')' indifferentemente dallo scopo, cosa che è stata corretta nelle versioni in allegato alla presente relazione.

## Domanda 4

La domanda 4 chiedeva di trovare sottostringhe simili ad una data stringa. L'approccio usato è stato quello di iterare sulla prima stringa, per tutti i possibili punti di inizio di una sottostringa simile. Il programma, utilizzando un ciclo *for* annidato, conta quindi tutti i caratteri uguali della sottostringa e della stringa da ricercare, e se questo numero risulta essere maggiore del minimo per essere considerata simile, ritorna l'indice di inizio della sottostringa.

Nel ciclo *for* esterno è stata eliminata la prima condizione, in quanto superflua perché già verificata dalla seconda.

È stato commesso un errore della condizione di uscita del *for* interno, in quanto la variabile *j* non parte da zero. La condizione è stata quindi modificata da "*j < strlen(str2)*" a "*j < i + strlen(str2)*".

## Domanda 5

Il primo errore riscontrato nella revisione del codice consiste nel non aver considerato il parametro intero *n* come dimensione della matrice. Questo non ha comunque cambiato il funzionamento dell'algoritmo, in quanto tutti i cicli sono stati parametrizzati in funzione della costante *DIM*, quindi è stato sufficiente cambiare *DIM* con *n* per risolvere il problema.

L'approccio utilizzato è stato quello di generare un vettore di float che contenesse le medie di ogni diagonale. Si è deciso di separare il calcolo di queste ultime in due fasi distinte. Nella prima sono state calcolate le medie delle diagonali inferiori e della diagonale principale, nella

seconda le medie delle diagonali superiori. Per fare ciò è stato utilizzato un ciclo *while* che itera sulle coordinate del punto di inizio di ciascuna diagonale, e un ciclo *for* annidato che calcola la sommatoria degli elementi sulla diagonale, incrementando ogni volta sia la coordinata *i* che la coordinata *j*. Per generare il doppio incremento all'interno delle condizioni del ciclo, è stata utilizzata un *and* logico, dato che si è supposto che l'incremento di una variabile venga sempre valutato come *true*. In realtà ciò non avviene nel caso limite in cui la prima variabile valga zero prima dell'incremento. Per ovviare a tale problema, è sufficiente cambiare il primo incremento in *++i*, in quanto avendo sempre valori delle coordinate maggiori o uguali a zero, il valore della variabile subito dopo l'incremento è sempre strettamente maggiore di zero. In realtà il problema è più facilmente risolvibile sapendo che all'interno costruito del *for* è possibile inserire più incrementi separati da virgole.

Un miglioramento netto dell'utilizzo della memoria si sarebbe potuto avere sfruttando un approccio on-line, cioè evitando l'utilizzo del vettore somme e valutando di volta in volta quale fosse la somma maggiore. In allegato, oltre la versione corretta della risoluzione originale, una versione ottimizzata con le precedenti considerazioni.

## Domanda 6

L'esercizio chiedeva di elaborare dati provenienti da un file di lunghezza incognita, ciò ha reso impossibile il salvataggio dei dati del file in una struttura dati senza dover imporre un numero massimo di righe del file. Si è quindi preferito l'approccio on-line, elaborando riga per riga il file.

La struttura dati utilizzata è stata dunque una matrice quadrata definita staticamente di dimensione sufficiente per contenere il piano considerato. La matrice è stata inizializzata al valore nullo per ogni suo elemento.

Per ogni riga letta dal file, l'algoritmo incrementa di uno i valori della matrice agli indici corrispondenti al rettangolo. In questo passaggio è stato commesso un errore in quanto il singolo ciclo *for* scritto va a incrementare di uno il valore degli elementi sulla "diagonale" del rettangolo, fermandosi quando uno dei due bordi (superiore o destro) viene raggiunto. È invece necessario utilizzare due cicli *for* annidati per incrementare l'intera area del rettangolo considerato. Questo errore è stato corretto nella versione allegata alla presente relazione.

Infine, per calcolare l'area totale delle regioni che si trovano all'intersezione tra due o più rettangoli, è stato sufficiente scorrere con due cicli *for* l'intera matrice e contare gli elementi che avessero valore strettamente maggiore di 1.

Il file utilizzato per la lettura dei dati è stato chiuso utilizzando la funzione *fclose*, in quanto si è supposto non dovesse essere più utilizzato nel main.