

PROVA PRATICA DI INFORMATICA & LAB. PROGRAMMAZIONE
20 dicembre 2021

Nome: _____ Cognome: _____ Matr: _____ Postazione _____

Scrivere un programma in linguaggio C (chiamare il progetto con la propria <matricola>) che abbia il comportamento descritto nel seguito. Il tempo a disposizione è di **120 minuti**. Al termine del tempo gli elaborati salvati su **Z:** verranno raccolti automaticamente dal sistema di laboratorio. Eventuali documenti sono disponibili in **P:\Bertozzi**, si consiglia di usare **wordpad** per leggere i file di testo.

Per la realizzazione del programma non utilizzare array o matrici allocate staticamente se non indicato diversamente nel testo. L'uso di variabili globali o numeri magici sarà inoltre sanzionato.

In enigmistica, l'intarsio è uno schema che partendo da due parole A e B ne "intreccia" le lettere generando una frase C.

Ad esempio da CANE e MICIO posso ottenere CAMICIONE. Indicando con 'a' le lettere provenienti dalla parola A e con 'b' quelle prese da B, la parola C è ottenuta tramite lo schema "aabbbbaa". Si noti come nella parola C le lettere mantengono l'ordine relativo che avevano nelle parole di origine.

L'algoritmo che genera lo schema risolutore è ben noto in letteratura e sfrutta una struttura a stack.

Si sviluppi un programma in C che:

1. apra e legga riga per riga un file "intarsi.txt" che su ogni riga contiene tre stringhe separate da ','. Le prime due stringhe non hanno mai spazi mentre la terza li può avere. Ad esempio:
GRADINO,AUTO,GARA DI NUOTO
Si ipotizzi una lunghezza massima per le tre parole di 50, 50 e 100 caratteri rispettivamente.
2. Ipotizzando che le tre stringhe lette siano rispettivamente le parole A e B e la frase C calcolare lo schema di intarsio che trasforma A e B in C. A tal fine sfruttare il seguente algoritmo:
analizzare le parole A, B e C lettera a lettera partendo dalle prime lettere e:
 - a. se la lettera in esame di C è uno spazio inserire nella sequenza risolutiva uno spazio e "avanzare" nell'analisi di C
 - b. se la lettera in esame di C è uguale alla lettera in esame di A ma diversa da quella di B porre nella sequenza risolutiva 'a'
 - c. se la lettera in esame di C è uguale alla lettera in esame di B ma diversa da quella di A porre nella sequenza risolutiva 'b'
 - d. se la lettera in esame di C è uguale alla lettera in esame di A ma anche a quella di B mi trovo di fronte ad un "bivio"Memorizzo in opportuna struttura a stack il punto in cui mi trovo ovvero, ad esempio, gli indici utilizzati per scorrere le tre parole.
Pongo nella sequenza risolutiva 'a' come se stessi scegliendo la lettera della parola A.

- e. se la lettera in esame di C è diversa da quelle in esame di A e B allora prelevo dalla struttura a stack l'ultimo "bivio" inserito e riparto da quel punto ponendo nella sequenza risolutiva 'b'
- f. se restano ancora lettere da considerare sia in A che in B ripetere dal punto (a)
- g. se restano ancora lettere da esaminare in A si inseriscano nella sequenza risolutiva tante 'a' quante sono le lettere rimanenti
- h. se restano ancora lettere da esaminare in B si inseriscano nella sequenza risolutiva tante 'b' quante sono le lettere rimanenti

Per tutti i punti precedenti si fa presente che ogni qual volta si inserisce nella sequenza risolutiva una 'a' occorre passare a considerare la lettera successiva di A e C. Analogamente quando si inserisce una 'b'.

Al termine del calcolo stampare le parole e la sequenza risolutiva.

Esempio di esecuzione:

Lo schema che da CANE e MICIO mi porta a CAMICIONE e' aabbbbbbaa

Lo schema che da GRADINO e AUTO mi porta a GARA DI NUOTO e' abaa aa ababb

Lo schema che da SARTO e PARIA mi porta a SPARATORIA e' abaabaabbb

Lo schema che da PIRA e RIME mi porta a PRIMIERA e' abbbabaa

Lo schema che da BRACA e ORCI mi porta a BORRACCIA e' ababaabba

Lo schema che da PROTO e ALTI mi porta a PARLOTTIO e' ababaabba

Lo schema che da SCORRI e OCRERA mi porta a SOCCORRERAI e' ababaabbbbaa

Lo schema che da CASCAO e ADAGIO mi porta a CASA DA GIOCO e' aaab bb bbbaaa

Si noti come nei primi due intarsi l'algoritmo proposto non necessiti di sfruttare bivi.