



Esercitazione di laboratorio n. 1

(Caricamento sul portale entro le 23.59 del 03/11/2017 dell'es. 3 e di uno tra gli es. 1 e 2)

Esercizio n. 1: Operazioni su insiemi

Competenze: manipolazioni di dati scalari e vettoriali

Categoria: problemi numerici iterativi su dati vettoriali (*Dal problema al programma: 3.1,4.1*)

Siano dati due vettori di interi, $V1$ e $V2$, inseriti da tastiera, di lunghezza $0 < V \leq 20$.

I due vettori possono avere lunghezze diverse.

Si scriva un programma in linguaggio C per rendere disponibili le seguenti operazioni:

- unione insiemistica tra i due vettori
- differenza insiemistica tra i due vettori
- intersezione insiemistica tra i due vettori

Esempio:

Sia il contenuto dei due vettori il seguente:

$V1 = \{1, 2, 1, 7, 8, 10\}$

$V2 = \{2, 1, 5, 6, 3, 3\}$

Allora:

$U = \{1, 2, 3, 5, 6, 7, 8, 10\}$

$D = \{3, 5, 6, 7, 8, 10\}$

$I = \{1, 2\}$

Esercizio n. 2: manipolazione di matrici

Competenze: lettura/scrittura di file, manipolazioni di matrici, costrutti iterativi.

Classificazione: problemi di selezione iterativi (*Dal problema al programma: 3.4.2, 4.5.2*)

Esercizio da 2 punti, parte di programmazione semplificata, appello del 18/09/2017.

Si scriva una funzione caratterizzata dal seguente prototipo:

```
int mat_search (char mat[20][20], int r, int c, char s[20]);
```

La funzione riceve una matrice di caratteri mat e una stringa s . La funzione conti quante volte la stringa appare in orizzontale o in verticale, in entrambi i sensi di lettura, nella matrice. Nell'esempio seguente la stringa foo compare 3 volte. Si osservi che sono lecite sovrapposizioni parziali tra stringhe.



	0	1	2	3	4
0	x	f	o	o	x
1	y	o	x	z	f
2	x	o	2	f	o
3	g	4	x	a	o

Esercizio n. 3: Codifica di testo con dizionario

Competenze: lettura/scrittura di file, manipolazioni di testi, ricerca in tabelle di nomi/stringhe

Categoria: problemi di elaborazione testi mediante stringhe (Dal problema al programma: 4.4.3)

Un file (`src.txt`) contiene un testo composto da un numero indefinito di righe, di lunghezza massima 200 caratteri ognuna. Un secondo file (`sost.txt`) contiene un elenco di coppie intero/stringa. Il file `sost.txt` è organizzato come segue:

- sulla prima riga è presente un numero intero e positivo S (≤ 30), che indica il numero di possibili sostituzioni presenti nel file
- seguono S coppie `<int> <nuovo>` a rappresentare le sostituzioni possibili, dove `<int>` è un numero intero positivo e `<nuovo>` è una stringa.

Lo scopo del programma è di codificare il primo file di testo andando a sostituire opportunamente alcune delle parole al suo interno sulla base dei contenuti del secondo file. Il risultato della compressione sia salvato su un terzo file (`dest.txt`), mantenendo la formattazione originale del testo.

La regola di sostituzione è: il programma deve contare la lunghezza di ogni parola incontrata nel file `src.txt`. Se esiste una coppia `<int> <nuovo>` nel file `sost.txt` dove `<int>` coincide con la lunghezza della parola corrente, allora il programma deve procedere alla sostituzione. Si assuma, nel caso in cui sia possibile sostituire, non via siano altre alternative associate al medesimo intero.

Esempio:

Il contenuto del file `src.txt` è:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Il contenuto del file `sost.txt` è:

5
3 ex
4 post
5 facto
8 justitia
9 memento

Il file di uscita `dest.txt` conterrà:



**POLITECNICO
DI TORINO**

03MNO ALGORITMI E PROGRAMMAZIONE CORSO DI LAUREA IN INGEGNERIA INFORMATICA A.A. 2017/18

facto facto facto ex post, consectetur adipiscing post, ex do
eiusmod tempor incididunt ut labore et dolore facto aliqua. Ut
post ad facto veniam, post nostrud exercitation ullamco laboris
post ut aliquip ex ea commodo memento.