

Corso di Sistemi Operativi e Reti, corso di Sistemi Operativi – Prova Pratica del 21 Febbraio 2013

1. **PER GLI STUDENTI DI SISTEMI OPERATIVI E RETI:** è necessario sostenere e consegnare **entrambi** gli esercizi. Sarà attribuito un **unico** voto su tutta la prova.
2. **PER GLI STUDENTI DI SISTEMI OPERATIVI:** si può sostenere solo uno dei due esercizi se si è già superata in un appello precedente la corrispondente prova. Il tempo a disposizione in questo caso è di **2 ORE**.

Troverete sul vostro Desktop una cartella chiamata "CognomeNomeMatricola" che contiene la traccia dell'elaborato ed eventuali altri file utili per lo svolgimento della prova. Ai fini del superamento della prova è indispensabile rinominare tale cartella sostituendo "Cognome" "Nome" e "Matricola" con i vostri dati personali. Ad esempio, uno studente che si chiama Alex Britti ed ha matricola 66052 dovrà rinominare la cartella "CognomeNomeMatricola" in "BrittiAlex66052".

Non saranno presi in considerazione file non chiaramente riconducibili al proprio autore. E' possibile caricare qualsiasi tipo di materiale didattico sul desktop nei primi 5 minuti della prova.

Si consiglia di salvare SPESSO il proprio lavoro.

ESERCIZIO 1 (Linguaggi di scripting)

Si vuole realizzare uno script perl `paroliere.pl` in grado di individuare, a partire da un insieme di lettere a disposizione, se l'insieme delle parole suggerite come possibili combinazioni contiene delle soluzioni valide ed attribuire a ciascuna il corretto punteggio sulla base di una tabella di valutazione.

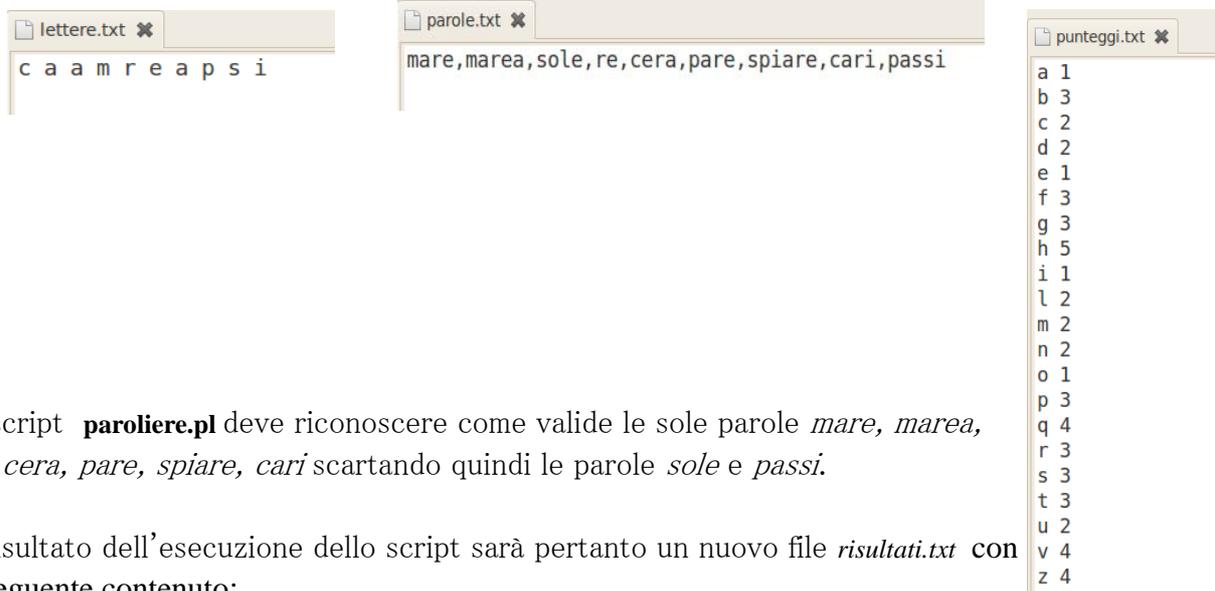
Più precisamente, lo script `paroliere.pl` riceve come input, da linea di comando:

- Il nome di un file, (es. `lettere.txt`), contenente un insieme di lettere a disposizione. Ciascuna lettera è separata da uno spazio;
- Il nome di un secondo file, (es. `parole.txt`), contenente un elenco di parole, separate da una virgola.

Lo script deve:

1. verificare per ciascuna parola contenuta nel file delle `parole.txt`, se è valida o meno, e nel caso in cui lo sia calcolare il suo punteggio.
Una parola del file `parole.txt` è valida rispetto all'insieme di lettere contenute nel file `lettere.txt` se è possibile costruirla utilizzando un sottoinsieme o tutte le lettere del file `lettere.txt`. Ciò significa che:
 - ciascuna lettera che compone la parola deve essere presente all'interno dell'insieme di lettere a disposizione
 - è possibile utilizzare ciascuna lettera 1 sola volta, ovvero, nel caso di parole contenenti ripetizioni di una stessa lettera, questa deve essere presente altrettante volte all'interno dell'insieme di lettere a disposizione in `lettere.txt`
 2. calcolare per ciascuna parola valida il suo punteggio rispetto ad una tabella di riferimento. La tabella di riferimento è fornita nel file `punteggi.txt` che fornisce su ciascuna riga un valore numerico attribuito a ciascuna lettera dell'alfabeto.
Il punteggio di una parola si ottiene sommando i punteggi attribuiti alle singole lettere che la compongono.
1. scrivere su un nuovo file `risultati.txt` i punteggi ottenuti per le parole valide, oppure nel caso in cui nessuna delle parole fornite siano valide, stampare a video una notifica per l'utente indicando che l'insieme fornito non presenta soluzioni valide.

Ad esempio, dati in input da linea di comando i file `lettere.txt` e `parole.txt`, come riportati di seguito, e considerando di avere una tabella di punteggi come quella che segue nel file `punteggi.txt`:



The image shows three text editor windows. The first window, titled 'lettere.txt', contains the characters 'c a a m r e a p s i'. The second window, titled 'parole.txt', contains the words 'mare, marea, sole, re, cera, pare, spiare, cari, passi'. The third window, titled 'punteggi.txt', contains a table of scores for each letter from 'a' to 'z':

a	1
b	3
c	2
d	2
e	1
f	3
g	3
h	5
i	1
l	2
m	2
n	2
o	1
p	3
q	4
r	3
s	3
t	3
u	2
v	4
z	4

lo script `paroliere.pl` deve riconoscere come valide le sole parole *mare*, *marea*, *re*, *cera*, *pare*, *spiare*, *cari* scartando quindi le parole *sole* e *passi*.

Il risultato dell'esecuzione dello script sarà pertanto un nuovo file `risultati.txt` con il seguente contenuto:

```
mare 7
marea 8
re 4
cera 7
pare 8
spiare 12
cari 7
```

ESERCIZIO 2 (Programmazione multithread)

Si scriva un programma (Java o C++) che, avendo a disposizione un processore con N core, riceve in input una stringa di testo T , una stringa S e un intero K , e restituisca in output *true* se la stringa S occorre in T almeno K volte, *false* altrimenti. Si assuma che T sia interamente disponibile in memoria, sia lungo almeno 500MB, e che sia necessario ottimizzare il programma in termini di efficienza sfruttando gli N core a disposizione. Si può assumere che S sia una stringa relativamente breve (10 caratteri max).

E' consentito (ma non obbligatorio), fare uso delle funzioni di ricerca su stringhe disponibili nel JDK di Java 6, e di qualsiasi altra funzione predefinita a disposizione. Non saranno valutate versioni sequenziali del programma.