

Laboratorio di Linguaggi di Sistema – a.a. 2005/2006 – Progetto finale II (base)

Il problema

Si vuole realizzare un semplice *programma di traduzione automatica* di testi. Il programma deve essere in grado di tradurre un file di testo dall'Inglese all'Italiano, come meglio specificato nel seguito. Il programma deve essere invocato con la seguente sintassi:

```
traduttore [-d dizionario] [file1] ... [filen]
```

Il programma deve inviare sullo standard output la versione tradotta del contenuto dei file indicati sulla riga di comando. Se sono presenti più file, il programma non deve inserire alcun elemento di separazione fra la traduzioni dei vari file. Se non è presente alcun file, il programma deve leggere il testo da tradurre dallo standard input. Se uno o più dei file indicati non sono leggibili, il programma li deve ignorare e passare oltre (segnalando però l'errore con un appropriato messaggio su standard error).

Formato del file di dizionario

Per effettuare la traduzione, il programma deve usare un file di dizionario, con il seguente formato: ciascuna voce inizia con la parola inglese, seguita nelle righe successive dalle possibili traduzioni italiane, precedute da un carattere di tabulazione. Ciascuna riga di traduzione contiene una o più possibili traduzioni, separate da virgola. La voce è terminata da una riga vuota. Per esempio:

```
a
    a, dentro

abacus
    abaco
    pallottoliere
    abaco, pallottoliere

abandon
    abbandonare

abandonment
    cessione

abashed
    timido

...
```

Il traduttore deve leggere il testo in ingresso, separarne le parole, e cercare ogni parola nel dizionario, sia nella versione originale che tutta in minuscolo (*suggerimento: si consiglia fortemente di caricare il dizionario in memoria, tramite opportune strutture dati, e poi consultare la versione in memoria, non quella su disco*). Se la parola è presente, deve scegliere di volta in volta a caso una delle traduzioni possibili; altrimenti, se la parola inglese termina per “s”, “es”, o “ies” (suffissi plurali) deve cercare nel dizionario la parola corrispondente in cui il suffisso plurale è sostituito da “”, “”, o “y” rispettivamente. Per esempio, “pennies” -> “penny”, “churches” -> “church”. Analogamente, se la parola termina per “ed” o “ing” (suffissi verbali) deve cercare la corrispondente radice senza il prefisso. Per esempio, “playing” -> “play”, “abandoned” -> “abandon”. *Avanzato**: lo studente può implementare altre riduzioni tipiche della lingua Inglese, a discrezione. Se in nessuno di questi casi la parola viene trovata, il programma deve stampare il termine inglese originale al posto della traduzione. In ogni caso, la formattazione e i simboli di punteggiatura devono essere conservati. *Avanzato****: lo studente può tentare di applicare al termine italiano le riduzioni corrispondenti ai suffissi rimossi dal termine inglese. Per esempio, “abandoning” -> “abandon” -> “abbandonare” -> “abbandonando”.

Se sulla riga di comando viene indicata l'opzione -d, il traduttore deve consultare il dizionario indicato dall'opzione; altrimenti il nome di default da usare è `ing-ita.txt`. Una copia del dizionario `ing-ita.txt` può essere scaricata all'URL

<http://circe.di.unipi.it/~gervasi/LLS05/ing-ita.txt>

Tempi e modalità di consegna

L'elaborato deve essere consegnato **improrogabilmente** entro il 23 giugno 2006 e deve essere costituito da una stampa del codice sorgente, adeguatamente formattato e commentato, da una stampa con esempi di esecuzione, e da una breve relazione scritta (2-3 pagine) che descrive il progetto stesso. La versione a stampa può essere consegnata direttamente al docente, oppure depositata presso il centralino del Dipartimento di Informatica. Tutto il materiale deve essere anche inviato via email al docente (gervasi@di.unipi.it) contestualmente alla consegna della parte cartacea.

Altre informazioni

Il programma deve gestire correttamente le situazioni eccezionali, quali ad esempio mancanza di memoria, impossibilità di accedere a certi file in quanto non esistenti o non dotati dei diritti necessari all'operazione, ecc. Gli eventuali errori devono essere segnalati all'utente attraverso messaggi su standard error. In nessun caso il programma deve inviare su standard output materiale diverso dalla traduzione dei testi di ingresso.

Ovviamente, non è necessario che la traduzione sia perfetta (né sensata).

Esempio

Testo in ingresso:

```
An abandoned rabbit is better than an abandoned abacus, my dear.
```

Testo in uscita nel caso più semplice (si noti che alcune parole hanno più traduzioni possibili, quindi altre versioni sono possibili):

```
An abbandonare coniglio is migliore che an abbandonare pallottoliere, mio caro.
```

Testo in uscita nel caso siano implementate le funzioni Avanzate*:

```
a abbandonare coniglio is migliore che a abbandonare pallottoliere, mio caro.
```

Testo in uscita nel caso siano implementate le funzioni Avanzate***:

```
a abbandonato coniglio is migliore che a abbandonato pallottoliere, mio caro.
```