

# Corso di Linguaggi Dinamici

## Esercitazione 3 sul Python

**Esercizio 1.** Scrivere un programma `splitFully` che, richiamando ricorsivamente il comando `os.path.split`, separi tutte le componenti di un path del file system passatogli come parametro.

Ad esempio: `splitFully('/usr/local/bin')` → `('/', 'usr', 'local', 'bin')`

Tempo a disposizione: **15 minuti.**

**Esercizio 2.** Scrivere un programma `printDir` che visualizzi il contenuto completo di una cartella del file system memorizzata in una variabile `path`. Quindi, scriverne una variante `printTree` che visualizzi ricorsivamente anche il contenuto delle eventuali relative sotto-cartelle.

Tempo a disposizione: **35 minuti.**

**Esercizio 3.** Scrivere un programma `rotate` che implementi la “rotazione” delle versioni precedenti di un file del file system. In particolare, dato il percorso di un file di cui si vuole effettuare un backup, ad esempio “`web.log`”, la funzione controllerà l’esistenza del file `e`, se esistente, rinominerà il file “`web.log.1`”. Ad una seconda invocazione, il programma dovrà far spazio ad un’altra versione di “`web.log`” facendo ruotare le precedenti versioni, cioè, nell’esempio precedente, chiamando la nuova “`web.log.1`” e rinominando “`web.log.1`” in “`web.log.2`”.

Tempo a disposizione: **35 minuti.**

**Esercizio 4.** Scrivere un semplice programma per la gestione di un blog. In particolare, dato il nome del file contenente i post, `blogAdd` dovrà leggere il contenuto del file ed effettuarne l’`unpickling` in una lista. Quindi, dovrà leggere la data e l’ora di sistema `date_and_time`, richiedere un nuovo post all’utente (string `text`), aggiungere la coppia `(date_and_time, text)` alla lista ed effettuare il `pickling` della lista nel file. `blogPrint` leggerà invece il contenuto del file per effettuarne la stampa. Utilizzare i moduli `pickle` e `time` (ad es.: `date_and_time = time.ctime(time.time())` ).

Tempo a disposizione: **45 minuti.**