


Fondamenti di Informatica - Ingegneria Informatica – Prof. Maristella Matera

Seconda prova in itinere – 2 Febbraio 2015

Il tempo massimo a disposizione per svolgere la prova è di **2h e 15m**. È permessa la consultazione dei soli manuali dei linguaggi; è vietato utilizzare calcolatrici, telefoni, PC. **Il voto minimo per superare la prova è 18.**

Esercizio 1. Gestione dei File e Matrici (14 punti)

La soluzione di un cruciverba può essere memorizzata in un file organizzato nel seguente modo:

- La prima riga contiene due numeri interi: il **numero di righe** e il **numero di colonne** del cruciverba;
- Le righe successive corrispondono alle **righe del cruciverba**; le caselle nere sono codificate come asterischi.

Di seguito si riporta un esempio di schema di cruciverba memorizzato nel file:

```
5 6
SUGLI*
ODIATI
TIRI*N
TRI*CD
OE*POI
```

Definire un programma in C in grado di leggere dal file lo schema del cruciverba e scrivere in un secondo file, chiamato **parole.txt**, l'elenco delle parole, sia orizzontali, sia verticali, che hanno bisogno di una definizione (cioè le sole parole di lunghezza ≥ 2). Il file **parole.txt** dovrà memorizzare prima tutte le parole orizzontali (una per riga), quindi una riga vuota, e infine tutte le parole verticali (una per riga). Per esempio, dato lo schema del cruciverba riportato sopra, il file **parole.txt** dovrà memorizzare:

```
SUGLI
ODIATI
TIRI
TRI
CD
OE
POI
```

```
SOTTO
UDIRE
GIRI
LAI
IT
CO
INDI
```

Si scriva **un programma in C** che riceve come **argomento sulla linea di comando il nome del file** che memorizza lo schema del cruciverba; quindi scrive nel file **parole.txt** l'elenco delle parole di lunghezza ≥ 2 , così come illustrato nell'esempio precedente. **Si assuma che la dimensione massima di un cruciverba sia 20×20 .**

Il programma deve essere realizzato in modo modulare, **scomponendolo almeno nelle seguenti funzioni:**

1. La funzione **main** con passaggio come argomento della linea di comando del nome del file con lo schema del cruciverba **(3 punti)**;
2. Una funzione che legge il cruciverba dal file trasferendone il contenuto in un'opportuna struttura dati **(4 punti)**;
3. Una o più funzioni per la ricerca delle parole nella struttura dati costruita al punto precedente e scrittura nel file **parole.txt (7 punti)**.

Esercizio 2. Liste dinamiche (10 punti)

Sia data una lista dinamica che memorizza dati riguardanti le quotazioni in borsa di alcune aziende. Gli elementi della lista rispettano la seguente definizione di tipo:

```
typedef struct nodo{
    char nomeAzienda[41];
    float quotazione;
    float variazione;
    struct nodo * next;
} aziendaQuot;
```

****La lista è ordinata in base al nome dell'azienda.****

Si definisca una funzione in C che riceve come parametri la testa della lista, il nome di un'azienda e un valore per l'ultima quotazione in borsa dell'azienda, e cerchi nella lista il nodo riguardante l'azienda:

- **Se il nodo esiste**, la funzione aggiorna il valore della quotazione e memorizza inoltre la variazione (differenza) rispetto al valore precedente nel campo **variazione**.
- **Se il nodo non esiste**, lo crea e lo inserisce nella posizione opportuna della lista, rispettando l'ordinamento alfabetico. La variazione in tal caso sarà uguale a 0.

La funzione restituisce il puntatore alla testa della lista (che potrebbe essere modificato da un inserimento in testa).

Esercizio 3. Funzioni ricorsive (8 punti)

Sia dato un array di valori interi, tutti maggiori di zero. **Definire una funzione ricorsiva che conta le “vette”, cioè gli elementi dell'array che sono maggiori di tutti gli elementi a essi successivi.**

Per esempio, l'array {10, 3, 4, 5, 1, 2} contiene due vette, 10 e 5, quindi la funzione per questo array restituirebbe il valore 2. Se l'array è vuoto o contiene un solo elemento, allora la funzione restituisce il valore 0.

Esercizio 4. Python (3 punti)

Sia data la seguente funzione in Python:

```
def foo (n):
    return [ i**2 for i in range (n) ]
```

Dire cosa stampa:

```
print foo(5)
```