

**Fondamenti di Informatica - Ingegneria Informatica – Prof. Maristella Matera**

Appello del 4 luglio 2016

Il tempo massimo a disposizione per svolgere la prova è di **2 ore e 15 minuti**. 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. Codifica binaria (4 punti)

1. Esprimere il numero decimale -234.875 in codifica binaria secondo lo standard IEEE 754 (in singola precisione a 32 bit);
2. Ricavare il valore decimale del seguente numero in virgola mobile rappresentato secondo lo standard IEEE 754 a 32 bit: 0 10000100 000100010000000000000000.

Esercizio 2. Funzioni ricorsive (8 punti)

Si consideri un array a di caratteri lungo n , che contiene alcune parole separate da un trattino. **Si definisca una funzione ricorsiva che restituisce 1 se la prima parola nell'array è formata in ordine dai caratteri iniziali delle restanti parole, 0 altrimenti.**

Esempio: Sia $n = 22$;

Se il contenuto di a è "PERA-PANE-EGO-ROSA-ALA", la funzione produrrà il valore 1;

Se il contenuto di a è "PANE-PERA-ECO-NOIA-EGO", il risultato sarà 0 poiché la terza parola non comincia per A.

N.B.: è eventualmente possibile definire una versione iterativa della funzione. Tale soluzione sarà tuttavia valutata con un massimo di 3 punti.

Esercizio 3. Uso di File e Liste Dinamiche (16 punti)

Si vuole definire un interprete in grado di individuare errori nell'uso delle parentesi in un testo. Si supponga di dover controllare l'uso di tre tipi di parentesi, utilizzate per delimitare parti di testo:

- le parentesi tonde: "(" e ")"
- le parentesi quadre: "[" e "]"
- le parentesi graffe "{" e "}"

Si scriva un programma che:

- **Legge da un file di testo** una sequenza di caratteri da analizzare e **memorizza i singoli caratteri in una lista dinamica**
- Quindi **verifica se la sequenza di caratteri letta da file è "ben parentesizzata"**. Una stringa è ben parentesizzata quando sono soddisfatte le due condizioni seguenti:
 1. Per ogni parentesi aperta (di uno dei tre tipi possibili) deve esserci nel seguito della stringa una parentesi chiusa dello stesso tipo;
 2. Ogni parentesi chiusa deve essere stata aperta in precedenza con una parentesi dello stesso tipo.

Esempi di stringhe:

- La stringa "]xxx(y)[]" non è ben parentesizzata, perché non soddisfa nessuna delle due condizioni.

- “[xxx(y)[aa]]” non è ben parentizzata perché, pur soddisfacendo la condizione 1, non soddisfa la 2 a causa della prima parentesi quadra - che è chiusa senza che prima ce ne sia una aperta.
- La stringa “(xx[y]” non soddisfa la prima condizione, mentre la stringa “x+(y-z)+1}” non soddisfa la seconda condizione.
- Sono invece ben parentizzate le stringhe: “{xxxx[y{zz}]}ww”, “({x})”, “xyyyzz”. Sono quindi valide sia stringhe che non contengono parentesi, sia quelle che non rispettano l’annidamento delle parentesi, come nella stringa “({x})” in cui l’ordine di chiusura delle parentesi non corrisponde a quello di apertura, purché le parentesi siano aperte e chiuse rispettando le due condizioni.

Esercizio 4. Python (5 punti)

Dato il seguente script in Python:

```
#!/usr/bin/python
```

```
def f(l):  
    if len(l)>1:  
        return l[0]+f(l[1:])  
    else :  
        return l[0]
```

```
p = "nome cognome"  
s = f([ c +'a' for c in p if c in 'this is a mighty long string' ])  
print s
```

Dire cosa stampa lo script se la variabile **p** è inizializzata, invece che con la stringa generica “**nome cognome**”, con il proprio nome e cognome sempre separati da uno spazio.