

PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA
29/9/2017

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (4 punti – vecchio ordinamento (ling. C): 5 punti)

Esprimere in complemento a due il valore a **cinque** bit 01001. Verificare che il valore ed il relativo complemento corrispondano l'uno all'opposto dell'altro.

ESERCIZIO 2 (solo vecchio ordinamento (ling. C): 5 punti)

Date due variabili booleane A e B Indicare a quali funzioni booleane corrispondono le seguenti funzioni espresse attraverso tabelle di verità:

A	B	F1(A,B)	F2(A,B)	F3(A,B)
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

ESERCIZIO 3 (4 punti – vecchio ordinamento (ling. C): 5 punti)

Descrivere in modo chiaro e sintetico i principali componenti di un moderno calcolatore elettronico.

ESERCIZIO 4 (25 punti – vecchio ordinamento (ling. C): 18 punti)

(5 punti – VO: 4 punti) Scrivere un programma in linguaggio Python che legga da un file "input.txt" una sequenza di interi compresi tra 0 e 9, e scriva, su un file "output.txt", il numero di valori letti, il valore più frequente, ed il numero di occorrenze di tale valore.

Esempio:

File "input.txt"
0
1
1
0
5
9
4
1

Con l'esempio di sopra il file output.txt va scritto come segue:

File "output.txt"
8
1
3

Si decide di memorizzare tutti gli elementi letti da file in una lista (vettore) `lista_input`, e i tre valori richiesti in uscita come primo, secondo e terzo elemento di una seconda lista `lista_output`, e di implementare le seguenti funzioni:

(2 punti) Funzione `inizializza` con:

- parametro in ingresso: un intero
- parametro in uscita: una lista (v.o.: puntatore a vettore di interi)

Restituisce una lista di tanti elementi quanto il valore fornito in ingresso, con tutti gli elementi posti a 0.

(7 punti – VO: 4 punti) Funzione `calcola_occorrenze` con:

- parametro in ingresso: la lista letta da file
- parametro in uscita: una lista che rappresenta, per ogni indice `i`, il numero di occorrenze di `i` nella lista in ingresso.

Calcola per ogni elemento presente nella lista di ingresso il numero relativo di occorrenze e lo restituisce mediante un'altra lista.

(7 punti – VO: 5 punti) Funzione `trova_max` con

- parametro in ingresso: una lista di interi
- parametro in uscita: l'indice corrispondente alla posizione nella lista del massimo valore

Restituisce l'indice dell'elemento della lista in ingresso corrispondente al massimo valore.

(4 punti – VO: 3 punti) Funzione `scrivi_dati`

- parametro in ingresso: una stringa e una lista di interi
- parametro in uscita: -

Scriva su file, di nome `nomefile`, fornito in ingresso mediante stringa, i valori presenti nella lista di ingresso separati dal carattere "a capo".

Soluzione dell'esercizio 1

Il valore dato è 01001, che in decimale corrisponde a 9.

Ora invertiamo ciascun bit: 10110.

Infine, sommiamo 1: 10111.

Per verificare la correttezza del risultato, considerando che i valori vanno espressi a 5 bit:

$$(10111) = -2^4 + 2^2 + 2^1 + 2^0 = -16 + 4 + 2 + 1 = -9$$

Un altro modo per verificare che l'uno è l'opposto dell'altro è usare la definizione di complemento a due:

$$01001 + 10111 = 100000$$

Il bit più significativo in neretto va scartato, e la rappresentazione rimasta è quella dello zero.

Soluzione dell'esercizio 2

Le funzioni presentano le seguenti tabelle di verità:

A	B	F1(A,B)	F2(A,B)	F3(A,B)
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

Dall'analisi delle stesse si ottiene

$$F1(A,B) = (A * B)' \text{ (NAND)}$$

$$F2(A,B) = (A + B)' \text{ (NOR)}$$

$$F3(A,B) = A \oplus B \text{ (XOR)}$$

Si veda il capitolo 1 per ulteriori dettagli.

Soluzione dell'esercizio 3.

Vedi dispense del corso (Cap. 2).

Soluzione dell'esercizio 4.

```
#Funzioni richieste
def inizializza(n):

    return [0] * n

def calcola_occorrenze(lista):

    occorrenze=inizializza(10)

    i=0
    while i<len(lista):
        j=lista[i]
        occorrenze[j]=occorrenze[j]+1
        i=i+1

    return occorrenze

def trova_max(lista):

    imax=0

    i=1
    while i<len(lista):
        if lista[i]>lista[imax]:
            imax=i
        i=i+1

    return imax

def scrivi_dati(nomefile, lista):
    f=open(nomefile, "w")

    for elemento in lista:
        f.write(str(elemento)+"\n")

    f.close()

#Corpo del programma
f=open("input.txt", "r")
lista_input=[]
l=f.readline()
while l!="":
    lista_input=lista_input+[int(l)]
    l=f.readline()
f.close()

lista_occorrenze=calcola_occorrenze(lista_input)
val_massimo=trova_max(lista_occorrenze)
lista_output=[len(lista_input), val_massimo, occorrenze[val_massimo]]
scrivi_dati("output.txt", lista_output)
```