

PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSO DI LAUREA IN INGEGNERIA CHIMICA E INGEGNERIA MECCANICA
17/2/2017

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

ESERCIZIO 1 (5 punti)

- 1) (2 punti) Esprimere in base 10 il valore nella rappresentazione in complemento a due con cinque bit 11001.
- 2) (3 punti) Descrivendo in dettaglio i passi dell'algoritmo, calcolare il corrispondente valore di segno opposto e verificare la correttezza del risultato.

ESERCIZIO 2 (5 punti)

Date due variabili booleane A e B, indicare a quali funzioni booleane corrispondono le seguenti funzioni F1, F2, F3 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 (5 punti)

Spiegare in modo chiaro e sintetico in cosa consiste il protocollo TCP/IP.

ESERCIZIO 4 (18 punti)

(3 punti) Scrivere un programma in linguaggio Python (C) che legga da un file "input.txt" una sequenza di interi compresi tra 0 e 9, scriva, su un file "output.txt", il numero di valori letti, il valore più frequente (moda), 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 scelgano le opportune strutture dati (liste e dizionari nel caso di Python, vettori e tipi strutturati in C) per l'implementazione del programma in funzione dei vincoli posti dalle seguenti funzioni da implementare:

(3 punti) Funzione `leggi_valori` con:

- ingresso: nome del file da leggere
- uscita: sequenza di valori letti, numero di valori letti.

Restituisce in una opportuna struttura dati la sequenza dei valori letti da file ed il loro numero.

(6 punti) Funzione `calcola_occorrenze` con:

- Ingresso: sequenza di valori letti
- Uscita: sequenza di occorrenze corrispondenti ai valori letti

(4 punti) Funzione `trova_max` con

- ingresso: sequenza di valori;
- uscita: l'indice corrispondente al massimo valore

(2 punti) Funzione `scrivi_dati`

- ingresso: sequenza di occorrenze, numero di valori letti, valore del massimo occorrente
- uscita: non necessaria.

Scriva su file, di nome `"output.txt"`, il numero di valori letti, il valore del massimo occorrente, ed il numero di occorrenze corrispondenti. La funzione restituisce `True` se l'operazione è andata a buon fine, `False` altrimenti.

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

$$F1(A,B) = (A * B)'$$

$$F2(A,B) = (A + B)'$$

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

Soluzione dell'esercizio 3.

Vedi dispense, Cap. 2, pagg. 96-98.

Soluzione dell'esercizio 4.

Nella soluzione che proponiamo utilizziamo una lista per memorizzare i valori letti da file e un dizionario per le occorrenze: la chiave sarà il valore a cui è associata la relativa occorrenza. Esempio d[3] è il numero delle volte che è stato letto il valore 3.

```
def main():
    listav, numv =leggi_valori("input.txt")
    occorrenze=calcola_occorrenze(listav)
    max_occ=trova_max(occorrenze)
    if scrivi_dati(occorrenze,numv,max_occ):
        print "\nOperazione compiuta con successo.\n"
    else:
        print "\nOperazione fallita.\n"

def leggi_valori(nomefile):
    idf=open(nomefile,"r")

    n=0
    lista_v=[]
    linea=idf.readline()
    while(linea!=""):
        v=int(linea)
        lista_v=lista_v+[v]
        n=n+1
        linea=idf.readline()

    idf.close()
    return lista_v, n

def calcola_occorrenze(valori):
    d={}
    for valore in valori:
        if valore in d:
            d[valore]=d[valore]+1
        else:
            d[valore]=1
    return d

def trova_max(d):
    massimov=0
    massimac=-1

    for c in d:
        if d[c]>massimov:
            massimov=d[c]
            massimac=c

    return massimac

def scrivi_dati(d,n,massimac):
    if d:
        idf=open("output.txt","w")
        stringa=str(n)+"\n"+str(massimac)+"\n"+str(d[massimac])+"\n"
        idf.write(stringa)
        idf.close()
        return True

    return False
```