Università degli Studi di Cagliari
Corsi di Laurea in Ingegneria Chimica e Ingegneria Meccanica

FONDAMENTI DI INFORMATICA

http://people.unica.it/gianlucamarcialis

A.A. 2018/2019

Docente: **Gian Luca Marcialis**

PYTHON: ESERCIZI DI PROGRAMMAZIONE PARTE 3

Esercizio/ricordo: Lettura/scrittura su file formattato

- Scrivere un programma che legge un file (nome da tastiera) di numeri e li scrive in un altro file (nome da tastiera) incolonnati a destra e seguiti dal totale e dal valore medio.

Soluzione

```
#Chiede all'utente i nomi dei file di input e di output
inputFileName=input("Inserire il nome del file di input:")
outputFileName=input("Inserire il nome del file dove scrivere il risultato:")

#Apre il file di input e di output
inFile=open(inputFileName,"r")
outFile=open(outputFileName,"w")
```

```
#Legge i dati e scrive il file di output
total=0.0
count=0

line=inFile.readline()
while line != "":
    valore=float(line)
    outFile.write("%15.2f\n" % valore)
    total=total+valore
    count = count + 1
    line=inFile.readline()
```

```
#Scrive il totale e il valore medio
outFile.write("%15s\n" % "--------")
outFile.write("Total: %8.2f\n" % total)

avg=total/count
outFile.write("Average: %6.2f\n" % avg)

#Chiude il file
inFile.close()
outFile.close()
```

Uso delle funzioni

Scrivere un programma Python che, leggendo da tastiera un valore di angolo espresso in gradi, scriva a video il valore fornito, lo stesso valore in radianti, e il corrispondente valore del seno, con due cifre significative dopo la virgola.

Nella scrittura della soluzione si implementino tre funzioni con i seguenti requisiti:

- Funzione main() : avvia il programma
- Funzione convertiInRadianti(angolo): riceve in ingresso il valore angolare in gradi e lo converte in radianti
- Funzione seno(angolo): riceve in ingresso il valore di un angolo in radianti e restituisce il relativo seno, restituendo zero nel caso l'angolo corrisponda a pi-greca

Esempio:

```
Inserisci un valore angolare in gradi: 90

 Gradi    Radianti        Seno

 90.00        1.57        1.00
```

Soluzione

```
from math import sin,pi

def main():
        angologradi=int(raw_input("Inserisci un valore angolare in gradi:"))
        angoloradianti=convertiInRadianti(angologradi)
        senangolo=seno(angoloradianti)
        print("%10s %10s %10s\n" % ("Gradi", "Radianti","Seno" ))
        print("%10.2f   %10.2f   %10.2f\n"   %   (angologradi,   angoloradianti,
senangolo))

def convertiInRadianti(angolo):
        angolo=(angolo*pi)/180
        return angolo

def seno(angolo):
"""Restituisce il seno di un angolo"""
        rapporto=angolo/pi
        if int(rapporto)==rapporto:
                return 0.0
        else:
                return sin(angolo)

main()
```

Conversione binaria

- Scrivere un programma che, ricevendo da tastiera una stringa che rappresenti un numero binario in complemento a 2 sul numero di bit pari alla lunghezza stringa data, scriva su file «out.txt» la stringa data, il valore decimale corrispondente, ed il opposto, poi anch'esso tradotto in decimale.
- Il numero letto parte dal meno significativo al più significativo
- Per l'esecuzione del programma si implementino le seguenti funzioni:
  - Y=Converti(X): Restituisce il valore decimale corrispondente alla stringa X
  - Y=Opposto(X): Restituisce la rappresentazione binaria di -X in accordo con l'algoritmo del complemento a 2

Esempio di file «out.txt»

```
Stringa binaria inserita:
0101        ← copia la stringa inserita da tastiera
Il valore decimale corrispondente è :
-6          ← conversione binario/decimale
Il suo opposto in complemento a due è :
0110        ← 0101 → 1010 + 1 = 0110
Che corrisponde a :
6           ← conversione binario/decimale
```

Il programma in pseudo-codice

```
X = leggi da tastiera una stringa binaria
Xt = traduci X in una lista di interi
Y = converti(X)
Y2 = opposto(X)
Stampa su file X, Y, Y2
```

Traduzione top-down

```
#X = leggi da tastiera una stringa binaria
X=raw_input('Dammi una stringa binaria: \n')

#Xc=converti X in decimale
Xc = converti(X)

#Y=calcola il complemento a 2 di X
Y = opposto(X)

#Yc=converti Y in decimale
Yc =converti(Y)

#Stampa su file X, Xc, Y, Yc
stampa(X, Xc, Y, Yc)
```

Implementazione di converti

```
"""Data una lista binaria X, restituisce il valore decimale corrispondente secondo la regola del complemento a 2"""

def converti(X):
      n=len(X)-1
      v=0
      i=0
      for x in X[0:n]
            v = v +  int(x) * 2**i
            i = i + 1
      v = v - int(X[n]) * 2**n

      return v
```

Implementazione di `opposto`

```
"""Data una stringa binaria X, restituisce il corrispondente
complemento a 2 come stringa binaria"""

def opposto(X):
        n=len(X)
        X=list(X)     #converto in lista di caratteri binari
        S=""          #inizializza una stringa vuota S

        #Copia in L il negato dei singoli valori in X
        #Aggiungi 1
        #Converti la lista binaria L in una stringa S

        return S
```

Implementazione di opposto

```
def opposto(X):
      n=len(X)
      X=list(X)    #converto in una lista di caratteri
      S=""         #inizializza una stringa vuota S

      i=0
      r=1          #devo aggiungere 1
      while i<N:
            b = int(not X[i])
            #bc, r = somma e riporto di b con riporto
            #          precedente
            S = S + str(bc)
            i=i+1


      return S
```

Implementazione di somma e riporto

```
""" Somma e riporto di b con riporto precedente
"""

def somma_riporto(x, y):
    s= (x and (not y)) or ((not x) and y)  #XOR
    r= x and y                             #AND
    return int(s), int(r)
```

Implementazioni di `opposto`

```
def opposto(X):
      n=len(X)
      S=""          #inizializza una stringa vuota S

      i=0
      r=1           #devo aggiungere 1
      while i<n:
            b = int(not X[i]) #inversione del bit
            bc, r = somma_riporto(b,r)
            S += str(bc)
            i=i+1

      return S
```

Implementazione di `stampa`

```
def stampa(X,Xc,Y,Yc):
	outf=open("out.txt", "w")

	outf.write("Stringa binaria inserita:\n"+X)
	outf.write("Il valore decimale corrispondente è
:\n"+str(Xc))
	outf.write("Il suo opposto in complemento a due è
:\n"+Y)
	outf.write("Che corrisponde a :\n"+str(Yc))

	outf.close()
```

Homework

- Copiare passo in un unico file attraverso l’editor ogni funzione scritta nelle slide precedenti
- Ultimare con le istruzioni di script nella slide 10
- Avviare il codice

Esercizio: Generazione istogrammi

- E' dato un file contenente i voti di studenti che hanno l'ultimo appello dell'esame di Fondamenti di Informatica
- I voti spaziano tra 0 e 30, con un voto superiore a 30 equivalente al 30 e lode (31, 32,…)
- Si scrivano le seguenti funzioni Python:
  - `leggi_voti`: Legge il file dei voti restituendo una lista coi voti. Il nome del file è fornito come parametro di ingresso. Durante la lettura, vanno ignorati i voti inferiori a 18 e tutti i voti superiori a 30 vanno riportati a 30.
  - `calcola_frequenze`: Calcola a partire da una lista di voti fornita in ingresso le frequenze di ciascun voto e la restituisce in uscita come lista.
  - `mostra_istogramma`: Stampa su file «output.txt» l'istogramma delle frequenze di ciascun voto nella modalità che verrà spiegata nella prossima slide. La lista delle frequenze dei voti è ricevuta in ingresso (`fvoti`).
  - `main`: Avvia il programma richiedendo il nome del file coi voti da tastiera ed eseguendo le successive elaborazioni.

Esempio di file «output.txt» generato dalla funzione `stampa_frequenze`

```
18 ********
19 **
20 **************
21 *******
...
30 *********
```

- In altre parole il file deve presentare il voto seguito dallo spazio e dalla relativa frequenza proporzionale al numero di caratteri "`*`"

Impariamo qualcosa di nuovo

- Generazione di numeri casuali
  - Esiste una funzione generatrice di valori numerici interi (pseudo)casuali nella libreria `random`
    - La funzione è `randint`:
      - `randint(minvalue,maxvalue)` – genera un numero intero pseudocasuale compreso tra minvalue e maxvalue
- Generazione di grafici
  - Esiste una libreria di funzioni che ci permette di fare dei grafici a partire da valori contenuti nelle liste, `matplotlib`
  - Noi estrarremo alcune funzioni da una sottolibreria, `pyplot`

Utilizzo della funzione generatrice di valori casuali

- Scriviamo una funzione che anziché leggere da file i voti, genera la lista utilizzando la funzione citata in precedenza, ricevendo in ingresso il numero di voti da generare

```
from random import randint
def genera_voti(m):
    voti=[]
    i=range(0,m)
    for n in i:
        voti=voti+[randint(18,30)]
    return voti
```

Esercizio

- Aggiungere la funzione scritta nella slide precedente e correggere la `main()` in modo che l'utente possa scegliere tra voti letti da file o generati casualmente.

Utilizzo della libreria grafica

- Utilizzeremo la libreria `matplotlib.pyplot`, invocandola come al solito ma rinominandola come plt per motivi di facilità di lettura:

```
import matplotlib.pyplot as plt
```

- Notare l'utilizzo della parola-chiave **`as`** per rinominare la libreria nel modo desiderato

Funzioni utilizzate (prendere nota)

- `figure()`
  - Genera spazio per una nuova «figura»
- `bar(voti,fvoti)`
  - Prenderà il range dei possibili voti (`voti`), le relative frequenze (`fvoti`), e genererà l'istogramma (grafico a barre) delle stesse
  - La funzione dispone di altre opzioni (allineamento delle etichette di voto, colore dell'istogramma)
- `xlabel(stringa), ylabel(stringa)`
  - Assegna agli assi di riferimento le etichette associate alla `stringa` data
- `show()`
  - Mostra a video la figura generata

Funzione alternativa `mostra_istogramma`

- Utilizzando le funzioni nuove, scriviamo una funzione alternativa a quella già scritta:

```
def mostra_istogramma(fvoti):
    voti=range(18,31) #genera l'intervallo
    plt.figure() #nuova figura
    plt.bar(voti,fvoti) #genera istogramma
    plt.xlabel('Voti') #label asse x
    plt.ylabel('Frequenza') #label asse y
    plt.show() #mostra a video la figura
```

Diagrammi a torta

- Dall'istogramma vogliamo ora a generare un diagramma a torta. Per far questo, ci servono le seguenti funzioni:
  - `calcola_frequenze_relative`: calcola e restituisce le frequenze relative ricevendo in ingresso quelle assolute per voto (fvoti). Le frequenze relative si calcolano dividento la frequenza assoluta del voto i-esimo per la somma di tutte le frequenze assolute.
  - `pie(fvoti, labels=voti)`
    - Funzione della libreria rinominata come `plt`
    - Utilizzando il range dei voti ammissibili memorizzato in `voti`, genera il diagramma a torta desiderato associando un colore diverso ad ogni «fetta», ovvero ad ogni voto, la cui frequenza relativa è memorizzata in `fvoti`.
    - L'assegnazione interna nel passaggio dei parametri attraverso `labels=voti` è obbligatoria
    - Anche questo diagramma si mostrerà a video con la funzione `show()`

Possibile soluzione

```
def mostra_torta(fvoti):

    voti=range(18,31)
    plt.figure()
    plt.pie(fvoti, labels=voti)
    plt.show()
```

Per saperne di più...

- K.A. Lambert, *Programmazione in Python*, Apogeo (Maggioli), 2012.
- C. Horstmann, R.D. Necaise, *Concetti di informatica e fondamenti di Python*, Apogeo (Maggioli), 2014.