Esercitazione di Informatica, 25 novembre 2015

Account provvisorio:
Username: lezione
PWD: 123.ciao

Esercizi in Python

1. Debugging (ricerca degli errori)

Aprite il file trovaglierrori.py (il docente vi indicherà dove trovare questo file, sia nella directory di rete dei laboratori, sia nella pagina web del corso). Trovate qui di seguito il contenuto del file, lo script contiene degli errori, trovateli e correggeteli.

```
lato=5
raggio=10
PI=3.74
#nota: numero**2 vuol dire numero elevato al quadrato
print ("Un quadrato di lato: "+lato+" ha un'area di: "lato**2)
area = (PI*raggio)**2
print ("Una circonferenza di raggio :"+raggio+" ha un'area di: area")
print ("E ora prova tu, immetti un lato di un quadrato: ")
lato = input()
print ("E poi immetti il raggio di una circonferenza:")
raggio = inbut()
print ("Un quadrato di lato: "+lato+" ha un'area di: "+lato)
arena = (PI*raggio)*2
print ("Una circonferenza di raggio :"+raggi0+" ha un'area di: "area)
print ("Corretto?)"
```

2. Debugging (variable explorer)

Aprite il file comandi.py (il docente vi indicherà dove trovare questo file, sia nella directory di rete dei laboratori, sia nella pagina web del corso). Provare ad utilizzare il debugger step-by-step per rispondere alle domande presenti nei commenti.

3. Input/Output

Scrivete un programma che prenda da tastiera una serie di numeri (almeno 2) e una serie di stringhe di testo (almeno 2),

- Stampate a video ogni valore inserito (devono essere stampati tutti i valori immessi)
- Stampate il tipo di ciascun valore immesso da tastiera
- Concatenate le stringhe e stampate il risultato
- Eseguite alcune operazioni matematiche a vostra scelta sui valori numerici inseriti e stampate il risultato

4. Celsius-Fahrenheit

Per misurare la temperatura, nel mondo si usano due scale: la scala Celsius (o centigrada), la scala Fahrenheit (utilizzata nei paesi di cultura anglosassone). Le scale di temperatura sono in genere caratterizzate dalla temperatura di fusione del ghiaccio e dalla temperatura di ebollizione dell'acqua. Nella tabella seguente trovate i valori per le due scale appena citate:

Scala	Temp. fusione ghiaccio	Temp. ebollizione dell'acqua
Celsius	0	100
Fahrenheit	32	212

La formula di conversione da gradi fahrenheit a gradi celsius è la seguente: $C = (F - 32) / 1{,}8$ (F rappresenta la temperatura in gradi fahrenheit e C rappresenta la temperatura in gradi celsius).
Scrivete un algoritmo in Python che:

- richiede in input (da tastiera) una temperatura in gradi Celsius

- stampa a video la temperatura equivalente in gradi Fahrenheit
- Testate l'algoritmo con i valori presenti nella tabella qua sopra

5. Somma Sottrazione Moltiplicazione

Scrivete un programma che richiede in input (da tastiera) due numeri, stampa il risultato della somma, della sottrazione e della moltiplicazione dei due numeri (nota: devono essere stampati 3 risultati).

6. Velocità

Scrivete un programma che dati da tastiera la distanza in Km e la velocità in Km/h calcola il tempo di percorrenza della distanza.

7. Convertitore di valuta

Scrivete un programma che richieda da tastiera i nomi di due valute (es. euro, dollaro, yen, …), il tasso di cambio e la quantità della prima valuta da convertire. Il programma deve calcolare il controvalore nella seconda valuta della quantità della prima valuta.

8. Calcolo sconto

Scrivete un programma che richieda da tastiera le seguenti informazioni: il prezzo unitario di vendita di un bene, la quantità venduta e lo sconto da applicare. Il programma dovrà visualizzare il prezzo totale di vendita sia prima di applicare lo sconto sia dopo averlo applicato.

9. Calcolo delle radici di un'equazione di 2° grado

Scrivete un programma che richieda da tastiera i coefficienti di un'equazione di secondo grado (a,b,c dell'equazione $aX^2 + bX + c = 0$). Il programma dovrà stampare le radici (le soluzioni) dell'equazione.

10. Trova il massimo

Scrivete un programma che richiede in input due valori numerici (tra loro diversi) e stampa il maggiore dei due. Nota: non occorre che il programma verifichi che i due valori siano diversi.

11. Calcolatrice:

Scrivete un programma python che richiede in input da tastiera 3 valori: due operandi e l'operazione da svolgere sui due operandi. Il primo e il secondo valore immessi da tastiera rappresentano gli operandi dell'operazione. Il terzo valore immesso da tastiera rappresenta un'operazione da svolgere (per esempio potete stabilire che "1" corrisponde all'operazione di somma, "2" corrisponde all'operazione di sottrazione, e così via).
Le operazioni ammesse sono somma, sottrazione, moltiplicazione e divisione. Stampate a video il risultato dell'operazione scelta dall'utente applicata ai due operandi numerici.

12. Identificare i divisori

Scrivete un programma che richiede in input due numeri e analizzando tali numeri riesce ad individuare le seguenti situazioni:

- il primo numero è un divisore del secondo;
- il secondo numero è un divisore del primo;
- nessuno dei due casi precedenti.

Il programma dovrà visualizzare con un messaggio a video in quale delle tre situazioni i due numeri si vengono a trovare.
Nota: vi conviene utilizzare l'operatore % (modulo) che calcola il resto della divisione intera di due numeri.

13. Classificazione

Scrivete un programma in python che richiede un numero intero da tastiera. Il programma deve classificare il numero inserito in una delle seguenti classi:
numeri minori di 0: classe K
numeri da 0 a 10: classe A
numeri da 11 a 20: classe B
numeri da 21 a 30: classe C
numeri da 31 a 40: classe E
numeri da 41 a 50: classe F
numeri > 50: classe G
Il programma deve stampare a video sia il numero inserito da tastiera, sia la classe di appartenenza.

14. Riconoscimento triangoli

Leggete tre valori in input, rispettivamente a, b e c che rappresentano le lunghezze dei tre lati di un triangolo. Richiedete che le lunghezze vengano inserite dalla più piccola alla più grande, cioè a<=b e b<=c, valutare se possono essere i tre lati di un triangolo, e se sì deciderne il tipo (scaleno, isoscele, equilatero) e stamparlo a video. Occorre tenere presente che:

- tre segmenti rappresentano i lati di un triangolo se il lato maggiore è minore della somma degli altri due
- se i lati sono tutti uguali si tratta di un triangolo equilatero
- se solo due sono uguali si tratta di un triangolo isoscele

Consiglio: provate a disegnare il flow chart dell'algoritmo prima di iniziare a implementarlo in Python.

15. Prezzi dei CD

Una casa stampatrice di CD applica alle aziende discografiche la tariffa di 5€ a CD se ne vengono ordinate meno di 1000 copie. Se la casa discografica ordina 1000 o più CD, può scegliere tra queste due forme di sconto:

- Pagare i CD che eccedono i primi 1000 al prezzo di 4€ l'uno (i primi 1000 cd sono comunque tariffati a 5€ l'uno)
- Pagare tutti i CD 5€ l'uno, ma ogni 20 cd venduti oltre i primi 800, uno viene regalato (questa forma di sconto non si applica se l'ordine non è di almeno1000 cd). Per esempio se una casa discografica ordina 1000 cd, ne pagherà solo1000-10 (questi ultimi costituiscono i cd regalati, 1 ogni 20 calcolati su una base di 1000-800=200); l'acquirente pagherà quindi 990cd a 5€ l'uno.

Scrivete un programma che richiede in input il numero dei cd da acquistare e visualizza in output se è applicabile uno sconto e quale delle due forme di sconto è più conveniente per la casa discografica.

16. Calcolo delle tasse sul reddito

Scrivete un programma che richiede in input lo stipendio guadagnato in un anno da una persona e ne calcola le tasse da pagare applicando il sistema di tassazione ad aliquote progressive per scaglioni di reddito descritto qui di seguito:

- Sui primi 10000€ non si pagano tasse
- Tra 10'000 e 22'000€ l'aliquota applicata è del 20%
- Tra 22'000 e 50'000€ l'aliquota applicata è del 30%
- Tra 50'000 e 100'000€ l'aliquota applicata è del 40%
- Sopra i 100'000€ l'aliquota applicata è del 50%

Esempio, una persona che percepisce un reddito di 35000€ pagherà una tassa così calcolata: tassa=(10'000 * 0% + 12'000 * 20% + 13'000 * 30%), dove 10000 è la parte di reddito che rientra nello scaglione non tassato, 12000 è la parte di reddito che rientra nello scaglione tassato al 20% e 13000 è la parte di reddito che rientra nello scaglione tassato al 30%.

Suggerimenti: provate prima a descrivere l'algoritmo per mezzo di un flowchart; utilizzate i contatori per realizzare l'algoritmo.

17. Esercizio libero

Provate a scrivere in python un algoritmo di vostra scelta che usi delle istruzioni condizionali annidate.

18. Condizioni

Prendete i file: **somma.py**, **genere.py**. Nel codice troverete degli asterischi, scrivete al loro posto le parti di codice necessarie per eseguire correttamente i programmi. Nel dettaglio:

1) Lo script somma.py controlla se la somma tra due numeri immessi da tastiera è *inferiore ad una soglia accettabile (max 100)*.
2) Sostituisci agli asterischi una condizione e la sua ramificazione. Lo script genere.py controlla due stringhe immesse da tastiera e sulla base del valore della condizione booleana declina al maschile o al femminile la frase da visualizzare a video.
3) Prendete il file **abbonamento_atm.py**. Il programma è caratterizzato da due condizioni, la seconda è annidata, verifica se uno studente ha i requisiti per ottenere un abbonamento studenti ATM. *Nel codice vi sono degli errori*, eseguendo il debugging dello script *identificate e correggete gli errori*.

19. Ordine crescente

Scrivere un programma che dati 3 valori numerici in input restituisce come risposta se i numeri sono in ordine crescente, nella risposta i numeri devono essere visualizzati, ma non occorre ordinare i numeri.

20. Anno bisestile

Scrivere un programma dall'algoritmo per calcolare l'anno bisestile. La regola dice che sono bisestili gli anni divisibili per quattro, tranne quelli di inizio secolo non divisibili per 400.

21. Media, numerosità predeterminata

Scrivete un programma python che calcola la media di un insieme di numeri immessi da tastiera. Il programma deve richiedere le dimensioni dell'insieme (cioè di quanti numeri dovrà essere calcolata la media) in anticipo, cioè prima di richiedere a tastiera i singoli valori. Stampate a video il valore della media. Per risolvere l'esercizio si suggerisce di introdurre un contatore per tenere conto del numero di valori inseriti.

22. Media, numerosità non determinata a priori

Scrivete un programma python che calcola la media di un insieme di numeri immessi da tastiera. A differenza dell'esercizio precedente, il programma non deve richiedere inizialmente la quantità di numeri da inserire, ma deve iniziare subito a richiedere i numeri dei quali deve calcolare la media. Quando la sequenza di numeri termina, l'utente inserisce uno zero come segnale che la sequenza di numeri è finita. Il programma dovrà stampare a video quanti numeri sono stati inseriti e la loro media.

***

Esercitazione di Informatica- Esercizi in Python

Account provvisorio:
Username: lezione
PWD: 123.ciao

23. Medie in contemporanea

Scrivete un programma che richiede in input 5 numeri e calcola:

- la media dei numeri positivi (>=0)
- la media dei numeri negativi (<0)

Le due medie devono essere visualizzate a video. Nota: i 5 numeri non devono essere richiesti una seconda volta per calcolare la seconda delle due medie. Stampa le due medie. Suggerimento: utilizzate i contatori.

24. Medie in contemporanea con valori nulli

Account provvisorio:
Username: lezione
PWD: 123.ciao

Come l'esercizio precedente, tuttavia considerate anche il caso in cui non sia possibile calcolare una delle due medie (es. si sono inseriti solo numeri positivi o solo numeri negativi).

25. Sequenza di Fibonacci

Scrivete un programma che calcoli i primi 20 numeri della sequenza di Fibonacci. La sequenza di Fibonacci è una sequenza di numeri che inizia con i numeri 0 ed 1 e ogni numero successivo è la somma dei due numeri precedenti. Un esempio della sequenza di Fibonacci è: 0 1 1 2 3 5 8 13 21 …

26. Indovina il numero

Scrivete un programma che implementa il gioco descritto qui di seguito:
Il programma genera casualmente un numero intero compreso tra 1 e 100, l'utente deve cercare di indovinare il numero andando a tentativi. Ad ogni tentativo l'utente deve scrivere un numero a tastiera e il programma deve comunicare all'utente se il numero inserito è rispettivamente, più grande, più piccolo del numero da indovinare oppure se è stato indovinato il numero. Il computer deve andare avanti a chiedere numeri fino a quando l'utente non indovina il numero. Per generare numeri casuali, usate le seguenti istruzioni:

```
import random     # da inserire all'inizio del programma python, importa la
                  # libreria contenente le funzioni per generare casualmente il numero
number = random.randint(0,100) #assegna alla variabile number un numero intero casuale compreso tra 1 e 100
```

27. Tabellina delle moltiplicazioni

Scrivete un programma che stampi a video la tabellina della moltiplicazione, dei numeri compresi tra 1 e 10.
Il programma dovrebbe produrre un output simile al seguente:

```
   1    2    3    4    5    6    7    8    9   10
   2    4    6    8   10   12   14   16   18   20
   3    6    9   12   15   18   21   24   27   30
   4    8   12   16   20   24   28   32   36   40
   5   10   15   20   25   30   35   40   45   50
   6   12   18   24   30   36   42   48   54   60
   7   14   21   28   35   42   49   56   63   70
   8   16   24   32   40   48   56   64   72   80
   9   18   27   36   45   54   63   72   81   90
  10   20   30   40   50   60   70   80   90  100
```

Il programma deve utilizzare due cicli annidati (il ciclo più interno deve stampare i singoli numeri di una riga, il ciclo più esterno deve ripetere l'operazione di stampa per più righe). Consigli: nel comando print aggiungete una virgola finale, come nell'esempio: print(…), in questo modo l'output

della print successiva sarà stampato sulla stessa riga senza andare a capo (questo escamotage funziona solo con Python 2, non con Python 3).

28. Conta i numeri

Scrivete un programma python che calcola quanti numeri pari e quanti dispari sono presenti in un insieme di numeri immessi da tastiera. Il programma deve richiedere le dimensioni dell'insieme (cioè di quanti numeri dovrà essere composto) in anticipo, cioè prima di richiedere a tastiera i singoli valori. Stampate a video i due sotto-insiemi (quanti pari e quanti dispari) e la loro somma.

29. Sette e mezzo

Scrivete un programma python che implementa il gioco di carte "Sette e mezzo". Lo scopo del gioco è quello di realizzare il punteggio più alto possibile senza mai *sballare*, vale a dire senza superare il 7 e mezzo. Il programma svolge il ruolo del mazziere, e genera casualmente un numero intero (la carta) compreso tra 1 e 10. Le carte dall'1 al 7 valgono tanti punti quanto è il loro valore numerico (1 vale 1 punto, il 2 vale 2 punti, ecc.) le carte dall'8 al 10 valgono mezzo punto. Il programma genera una carta per volta e chiede all'utente se vuole continuare a giocare fino a quando l'utente non ha vinto o ha superato il 7 e mezzo (in tal caso ha perso).
Per generare numeri casuali, usate le seguenti istruzioni:

```
import random        # da inserire all'inizio del programma python, importa la
                     # libreria contenente le funzioni per generare casualmente il numero
carta = random.randint(1,10) #assegna alla variabile carta un numero intero casuale compreso tra 1 e 10.
```

30. Lista 1

Modificate l'esercizio 28 in modo da lavorare su una lista di numeri precaricata (la lunghezza della lista a vostra scelta) al posto dei valori immessi da tastiera e calcolate quanti numeri pari e quanti dispari ci sono.

31. Maggiore e minore

Scrivete un programma che a partire da una lista contenente 10 numeri già inseriti, cerca il minore ed il maggiore tra i numeri presenti. Per comodità potete copiare ed incollare nel vostro programma il seguente comando: li=[30, 18, 54, 90, 6, 150, 114, 162, 72, 78]

32. Moltiplicazione di una matrice riga per una matrice colonna.

Date due liste: r = [2, 4, 10, 22, 5] c= [10, -1, 5, 3, 12]
supponete che le due liste rappresentino due matrici, rispettivamente una matrice riga di dimensioni [1,5] ed una matrice colonna di dimensioni [5,1]. Scrivete un programma che calcola il prodotto delle due matrici. Il prodotto in questo caso è un intero pari alla somma dei prodotti degli elementi corrispondenti delle due liste. Il programma dovrà stampare alla fine il risultato trovato. Se non conoscete il calcolo matriciale chiedete al docente.

33. Controllo divisori

Scrivete un programma che crea due liste con 10 numeri interi l'una (scegliete voi se inserire i numeri da tastiera oppure inizializzare le liste con dei numeri direttamente dentro il programma). Il programma dovrà costruire una (terza) lista di dieci valori booleani, ognuno di questi dirà se il corrispondente elemento della seconda lista è un divisore intero del corrispondente elemento della prima lista (cioè se il resto della divisione dei due elementi è nullo). Il programma dovrà stampare poi le tre liste.

34. Dati su tuple

Scrivete un programma che richiede il cognome, il nome e l'età di una persona. Queste informazioni devono essere memorizzate in una tupla, la quale sarà aggiunta in coda ad una lista. Il programma

dovrà ripetere l'operazione di inserimento delle informazioni per 4 persone. Alla fine dell'inserimento il programma dovrà mostrare i dati delle persone che hanno "Rossi" come cognome.

35. Funzioni 1

Prendete l'esercizio Calcolatrice (1. Foglio), e modificatelo in modo che la parte del programma che opera come una calcolatrice sia il corpo di una funzione che viene richiamata dal nuovo programma che scriverete (scegliete voi cosa inserire nel corpo principale del programma per testare la funzione).

36. Funzioni 2

Prendete l'esercizio Classificazione (1. Foglio), e modificatelo in modo che la parte del programma che classifica un numero sia il corpo di una funzione che viene richiamata dal nuovo programma che scriverete (scegliete voi cosa inserire nel corpo principale del programma per testare la funzione).

37. Funzioni 3

Prendete l'esercizio Riconoscimento triangoli (1. Foglio), e modificatelo in modo che la parte del programma che riconosce un triangolo sia il corpo di una funzione che viene richiamata dal nuovo programma che scriverete (scegliete voi cosa inserire nel corpo principale del programma per testare la funzione).

38. Funzioni 4

Scrivete una funzione (nome della funzione: e_pari) che prende in ingresso un numero e restituisce vero o falso a seconda che il numero passato come parametro sia pari o dispari. Scrivete poi nello stesso file il corpo principale del programma che usa la funzione (scegliete voi cosa inserire nel corpo principale del programma per testare la funzione).

39. Funzioni 5

Prendete la funzione e_pari dell'esercizio precedente, salvatela in un file di libreria (nomefile: mialibreria.py), implementate nel file di libreria la funzione e_dispari (che restituisce vero o falso a seconda che il numero passato come parametro sia dispari o pari). Create un altro script python (es. calcoli.py) che importa ed usa le due funzioni di libreria.
Accertatevi che nel file mialibreria.py ci sia solamente l'implementazione delle due funzioni (e_pari, e_dispari) e nient'altro (non copiate il corpo principale dell'applicazione descritta nel punto precedente).

40. Ordinamento tramite Bubble Sort

Scrivere un programma che richiede all'utente di inserire 10 numeri da tastiera, li inserisce in una lista, ordina la lista in modo crescente (dal numero più piccolo al numero più grande) e stampa la lista ordinata.
Si consiglia, durante la stesura del programma, per fare le prove, di utilizzare la seguente lista di numeri, in questo modo, durante ogni esecuzione, non dovrete ogni volta inserire i 10 numeri:

li=[30, 18, 54, 90, 6, 150, 114, 162, 72, 78]

Per ordinare la lista, implementate l'algoritmo del bubble sort, (non utilizzate gli algoritmi già pronti delle librerie python). L'algoritmo del bubble sort è descritto qui di seguito:

- Confrontate il primo elemento della lista con l'elemento successivo, se non sono in ordine (cioè se il primo elemento è maggiore del secondo) scambiateli tra di loro. Se invece sono in ordine procedete al punto successivo senza fare niente
- Ripetete l'operazione di cui sopra tra il 2° e il 3° elemento, tra il 3° e 4° elemento e così via fino ad arrivare a confrontare il penultimo e ultimo elemento della lista
- Se si porta a termine un insieme di controlli (per insieme di controlli si intendono le operazioni descritte nei punti precedenti di confronto tra tutti gli elementi) senza effettuare neanche uno scambio, allora la lista è ordinata e il programma può terminare. Se invece durante un insieme di controlli avviene uno scambio, al termine occorre ripetere da capo l'insieme dei controlli (dal primo all'ultimo elemento) fino a che non si effettueranno più scambi. Quando non saranno più

necessari scambi, l'algoritmo di ordinamento avrà terminato. Vi conviene utilizzare una variabile flag per verificare se durante un controllo sono stati effettuati o meno degli scambi

41. Ordinamento per selezione

Scrivere un programma che richiede all'utente di inserire 10 numeri da tastiera, li inserisce in una lista, ordina la lista in modo crescente (dal numero più piccolo al numero più grande) e stampa la lista ordinata.
Si consiglia, durante la stesura del programma, per fare le prove, di utilizzare una lista di numeri precaricata, come quella riportata nell'esercizio precedente.
Per ordinare la lista, implementate l'algoritmo di ordinamento per selezione. L'algoritmo è descritto qui di seguito:

- Cercate nella lista la posizione dell'elemento più piccolo. Scambiate l'elemento più piccolo con l'elemento di indice 0 (il primo elemento della lista). In questo modo si sposta a sinistra l'elemento più piccolo della lista.
- Cercate l'elemento più piccolo tra gli elementi rimanenti (si tratta di tutti gli elementi della lista ad esclusione di quello di indice 0; detto in altro modo, la ricerca deve essere effettuata a partire dall'elemento di indice 1 fino all'ultimo elemento della lista). Scambiate l'elemento trovato con l'elemento di indice 1.
- Ripetete il procedimento descritto nei punti precedenti cercando l'elemento più piccolo a partire dalla posizione *i* fino alla fine della lista, e una volta trovato, scambiandolo con l'elemento in posizione *i*. Questa operazione va ripetuta per tutti i valori di *i* compresi tra 0 e la lunghezza della lista. Di fatto ogni volta si cerca l'elemento minimo tra quelli rimanenti e lo si sposta a sinistra. Quando si raggiunge la fine della lista, la lista è ordinata

42. Calcolo della moda

Scrivete un programma che richiede in ingresso una sequenza di numeri e li memorizza in una lista. La sequenza può essere composta da una quantità arbitraria di numeri, la sequenza termina quando l'utente inserisce il numero 0.
Il programma deve stampare a video la moda (il numero che appare più spesso, o a parità di numero di apparizioni, il numero più piccolo). Attenzione, questo esercizio è complicato. Consigli:

- Utilizzate un dizionario di contatori per contare la numerosità con cui appaiono i diversi numeri: per ogni numero di cui occorre contare la numerosità, nel dizionario si inserisce una coppia formata dal numero (chiave) e dal contatore associato (variabile associata alla chiave)
- Una volta terminato il conteggio delle numerosità, individuate il massimo tra le diverse numerosità
- cercate il numero (o i numeri perché possono essere più di uno) che hanno come numerosità la numerosità massima precedentemente individuata

43. Media e varianza di numeri generati casualmente

Scrivete un programma che generi 1'000 numeri casuali interi (di valore compreso tra 0 e 100) e li memorizzi in un dizionario. I numeri casuali devono essere inseriti nel dizionario per mezzo di una coppia chiave:valore dove la chiave è un numero progressivo che va da 0 a 999 e il valore è il numero casuale. Successivamente il programma deve calcolare media, varianza e scarto quadratico medio dei numeri generati casualmente presenti nel dizionario.
Per generare numeri casuali, usate le seguenti istruzioni:

```
import random     # da inserire all'inizio del programma python, importa la
                  # libreria contenente le funzioni per generare casualmente il numero
numero = random.randint(1,100) #assegna alla variabile numero un numero intero casuale compreso tra 1 e 100.
```

Ripetete lo stesso esercizio generando 10'000 numeri casuali.

44. Combinazioni di nomi e cognomi

Create una lista di 5 nomi. Successivamente create una lista di 5 cognomi. Create un dizionario nel quale saranno inserite delle coppie chiave:valore così costituite: la chiave è un numero progressivo, il valore è una tupla formata da un nome e un cognome. Nel dizionario dovranno essere inserite tante tuple quante sono le possibili combinazioni di nomi e cognomi.

45. Funzioni: calcolo della radice quadrata

Scrivete una funzione che accetta in ingresso due valori numerici (di tipo float) che chiameremo rispettivamente A e B. A rappresenta un numero del quale deve essere calcolata la radice quadrata (potete assumere che A sia un numero non negativo). Il valore numerico di B invece rappresenta la "precisione" con il quale dovrà essere calcolata la radice quadrata (vedi il proseguimento della spiegazione). Il programma che scriverete dovrà calcolare la radice quadrata di A, non utilizzando le funzioni matematiche già presenti in Python (cioè non potete utilizzare il metodo già visto a lezione), ma dovrà utilizzare un algoritmo scritto da voi. L'algoritmo che dovrete implementare è il seguente:
Utilizzate due variabili aggiuntive (es. top e bottom) che rappresentano l'intervallo nel quale può trovarsi la soluzione. All'inizio a bottom deve essere assegnato il valore 0 e a top deve essere assegnato il valore di A.
Ripetete più volte il gruppo di operazioni descritte qui di seguito:

- Calcolate il valore mediano tra top e bottom (es. tra 8 e 0 il valore mediano è 4; tra 10 e 2 il valore mediano è 6).
- Elevate il mediano al quadrato e confrontatelo con A, se il mediano al quadrato è superiore ad A, il mediano diventerà il nuovo valore di top, se invece il mediano al quadrato è inferiore ad A, il mediano diventerà il nuovo valore di bottom.

Come già detto, le variabili top e bottom rappresentano un intervallo nel quale si trova la soluzione cercata. Ripetendo più volte la sequenza di operazioni descritta sopra, i valori di top e bottom si avvicinano tra loro sempre più. Quando la differenza tra top e bottom diventa inferiore alla "precisione" (cioè al secondo valore fornito in input dall'utente) l'algoritmo deve terminare fornendo come risultato della radice quadrata il valore mediano calcolato sugli ultimi valori di top e bottom.

46. Trova i duplicati

Scrivere una funzione che riceve in ingresso una lista di interi (di lunghezza non nota a priori) e restituisce il numero di elementi duplicati (cioè quanti numeri sono presenti 2 o più volte). Per esempio nella lista [1, 5, 5, 5, 4, 4, 0, 8, 7, 7] i numeri duplicati sono 3 (cioè i numeri 5, 4, 7). Suggerimento: è più semplice individuare i duplicati dopo aver ordinato la lista (tutti i duplicati di uno stesso numero si vengono a trovare in posizioni adiacenti).

47. Differenza di due liste

Scrivete una funzione che accetta due liste di numeri interi come parametri in ingresso. La funzione dovrà restituire in uscita una lista formata da tutti i numeri presenti nella prima e non presenti nella seconda. Nella lista restituita dalla funzione, ogni numero deve apparire una sola volta. Per esempio, se si forniscono in ingresso le liste L1=[9, 1, 1, 3, 5, 4, 5] e L2=[1,3] la funzione dovrà restituire la lista [9, 4, 5]. Non è richiesto che la lista fornita in uscita sia ordinata. Suggerimento: si consiglia di usare il comando "in" per testare l'inclusione di un elemento in una lista.

48. Elenco dei divisori

Scrivete una funzione, che accetta come parametri due liste di numeri intere *la* ed *lb*, aventi entrambe lunghezza non nota a priori. La funzione deve restituire una lista che contiene gli elementi di *lb* che sono divisori di tutti gli elementi di *la*. Per esempio, se *la*=[12, 6, 36] e *lb*=[2, 3, 4, 5, 10, 11], la lista restituita dovrà essere [2, 3]. Se vi può essere d'aiuto potete implementare più di una funzione. Testate la vostra funzione richiamandola e passandogli alcuni valori.

49. Conteggio occorrenze sottostringa

Scrivete una funzione che, a partire da due parametri s1 ed s2 che memorizzeranno due stringhe la cui lunghezza non è nota a priori, restituisce un numero che indica quante volte la seconda stringa è presente all'interno della prima. Per esempio, se s1='axyzbxyzcxyz' e s2='xyz', la funzione dovrà restituire il numero 3. Se s2 non è presente all'interno di s1 il programma dovrà visualizzare il numero 0. Non potete utilizzare il metodo find() delle stringhe, in altre parole dovete implementare voi l'algoritmo (ignorate questa frase se non sapete che cosa è il metodo find).
Per vostra semplicità assumete (senza cioè la necessità di dover fare controlli) che la lunghezza di s1 sia maggiore di s2. Suggerimento: verificate, a partire da ogni carattere di s1 se il carattere stesso e i caratteri successivi corrispondono ai caratteri di s2. Testate la vostra funzione richiamandola e passandogli alcuni valori.

50. Sequenze di numeri uguali (almeno due consecutivi)

Scrivete una funzione che, a partire da un parametro che memorizza una lista di numeri interi di lunghezza non nota a priori, restituisca un numero corrispondente a quante sequenze di numeri uguali (con almeno 2 elementi) sono presenti all'interno della lista.
Per esempio se nf=[10, 1, 1, 1, 1, 5, 3, 3, 6] lo script deve visualizzare a video il numero 2. Nota bene: una sequenza di numeri uguali contigui deve essere contata una volta solo fino al raggiungimento del maggior numero possibile di elementi, nell'esempio precedente la sequenza 1,1,1,1 deve essere contata come una sequenza unica e non come due sequenze di 1,1. Testate la vostra funzione richiamandola e passandogli alcuni valori.

51. Split

Scrivete una funzione che accetta come parametro una stringa contenente delle parole separate da spazi e restituisce una lista i cui elementi sono le singole parole, ottenute utilizzando gli spazi per separare le parole della stringa.
Per esempio, se si passa alla funzione la stringa s='Tanti saluti al Signor Rossi', la funzione deve restituire la lista ['Tanti', 'saluti', 'al', 'Signor', 'Rossi']. Non potete utilizzare il metodo .split() delle stringhe, in altre parole dovete implementare voi l'algoritmo (se non sapete che cosa è il metodo .split() ignorate tranquillamente questa frase).
Per vostra semplicità, assumete che nella stringa passata come parametro non ci sia mai più di uno spazio consecutivo (non esistono cioè 2 o più spazi uno dietro l'altro). Testate la vostra funzione richiamandola e passandogli alcuni valori.

52. Conta temperature

Scrivete una funzione che accetta come parametri due liste di numeri interi: previsione e reale. Le due liste contengono rispettivamente i valori di temperatura giornaliera previsti dalle previsioni del tempo (la lista previsione) e i valori che sono stati registrati realmente (contenuti nella lista reale). Temperatura prevista e temperatura reale corrispondente hanno lo stesso indice nelle due liste. Quindi, le due liste hanno sempre lo stesso quantitativo di numeri.
La funzione, per ogni valore di temperatura presente nelle liste previsione e reale, deve calcolare quante volte tale temperatura appare nella lista previsione diviso quante volte tale temperatura appare nella lista $reale$. Un valore di temperatura, anche se appare più volte nelle due liste, deve essere considerato una volta sola.
I risultati dovranno essere restituiti tramite un dizionario di coppie chiave valore, dove la temperatura costituisce la chiave e il valore associato è il rapporto precedentemente descritto.
Per esempio, se previsione=[20,30,30,15] e reale=[20,20,30,18], la funzione dovrà restituire il dizionario {20:0.5, 30:2.0}.
Nel vostro script, assumete che le liste previsione e $reale$ siano già presenti in memoria (in altre parole, non dovete eseguire l'input). Testate la vostra funzione richiamandola e passandogli alcuni valori.

53. Scrittura su file

Scrivete un programma che richiede in ingresso cognome, nome e matricola di 5 studenti.
Il programma scrive i dati su un file di testo, separando con una virgola il cognome, il nome e la matricola e andando a capo dopo aver inserito i dati di una persona. I dati salvati su file dovranno apparire in questo modo:
nome1,cognome1,matricola1,
nome2,cognome2,matricola2,
nome3,cognome3,matricola3

54. Lettura e scrittura su file

Scrivete un programma che richiede in input una sequenza di numeri interi diversi da zero (terminati da uno 0 che comunica che l'immissione è finita). Il programma deve scrivere questi numeri su un file di testo, separandoli da una virgola. Scrivete un secondo programma che legge il file scritto dal programma precedente, carica i valori numerici in una lista e la stampa a video.

55. File CSV

Troverete nella cartella di rete condivisa (e dopo il termine della lezione anche nel sito web del corso) un file dal nome "dataset_persone.zip", il quale all'interno contiene il file dataset_cognomi_nomi_eta_comunenascita.txt. Quest'ultimo è un file in formato CSV (comma separated value) che contiene al suo interno informazioni su un insieme di persone. Ogni riga del file riporta i dati di una singola persona, le informazioni presenti in una riga sono nell'ordine: nome, cognome, genere, età, comune di nascita.
Qui sotto potete vedere un estratto del file CSV
FABIO;GILARDI;M;22;ALATRI
STEFANIA;LICCIARDELLO;F;12;MONTENERO SABINO
ANNALENA;MAINI;F;43;CASALUCE
Utilizzando i dati contenuti nel file, scrivete dei programmi in python in grado di fornire dei risultati utili per rispondere alle domande riportate qui di seguito.
Importante: leggete le note finali prima di iniziare gli esercizi.

A. Quantità totali e distinte per genere

Quante sono le persone descritte nel file? Quante sono le persone di sesso maschile, quante le persone di sesso femminile?

B. Fasce d'età

Tutte le persone descritte nel file hanno un'età compresa tra 1 e 59 anni. Date 6 fasce d'età (1-9, 10-19, 20-29, 30-39, 40-49, 50-59), individuate quale è la numerosità (quante persone fanno parte) di ogni fascia d'età.

C. Numerosità cognomi

Per ogni cognome presente, stampate a video la numerosità associata al cognome. Ogni cognome deve apparire a video una sola volta.

D. Cognome, nome, luogo di nascita più diffusi

Individuate il cognome più diffuso e la sua numerosità (la numerosità delle persone che hanno tale cognome); individuate il nome più diffuso e la sua numerosità, individuate il comune di nascita più diffuso e la numerosità. Nota: si tratta di 3 esercizi distinti.

Note

- Copiate il file .zip in locale ed estraete il contenuto nella directory in cui lavorerete

- Il file contenente il dataset è stato generato casualmente a partire dall'elenco dei nomi e cognomi più diffusi in Italia nel 2006 e dall'elenco dei comuni italiani (entrambi disponibili sul sito web dell'istat)
- Per svolgere gli esercizi, occorre aver letto/studiato le slide della lezione sui file, in modo particolare è fondamentale utilizzare i metodi "strip" e "split".

All'inizio, non vi conviene lavorare con il dataset completo dei dati. Createvi un file ridotto (per esempio con solo le prime 50 righe del file originale) e lavorate su quello. Quando avrete acquisito maggiore esperienza,

56. Esercizi a piacere

Svolgete delle analisi a vostro piacere sul dataset precedente

57. File CSV 2 (film)

Troverete nella cartella di rete condivisa (e dopo il termine della lezione anche nel sito web del corso) un file dal nome "catalogo_film.zip", il quale all'interno contiene il file catalogo_film.xls. Quest'ultimo è un file excel, dal quale dovrete estrarre il corrispondente file in formato CSV (comma separated value). Ogni riga del file riporta i dati di un singolo film, le informazioni presenti in una riga sono nell'ordine: codice, genere, titolo, regista, attori, prezzo.
Utilizzando i dati contenuti nel file, svolgete delle analisi a vostro piacere, per es. quanti sono i film venduti ad un certo prezzo, divisi per genere; quali film di un certo regista, ecc.

Note

- Copiate il file .zip in locale ed estraete il contenuto nella directory in cui lavorerete
- Il file contenente il dataset è stato generato casualmente a partire dall'elenco dei film di una videoteca
- Attenzione che alcuni campi (soprattutto nelle colonne: regista, attori) possono essere vuoti. Inoltre le stringhe che riportano il titolo e il genere possono essere interrotte, o con parole abbreviate, tenetene conto nelle analisi che svolgerete