

Lab 01

Elementi di Informatica e Programmazione

2023-10-27

Esercizio 1-1

Progettare il programma `twoNumbers.py` che chieda all'utente di inserire due numeri e ne visualizzi:

- la somma
- il prodotto
- il valore medio
- il valore massimo
- il valore minimo
- il valore assoluto della differenza

Soluzione

```
n1 = float(input("inserire un numero"))
n2 = float(input("inserire un altro numero"))

print("somma", n1 + n2)
print("prodotto", n1 * n2)
print("media", (n1 + n2) / 2)

# Massimo e minimo possono essere fatti con uno statement condizionale
if n1 > n2:
    print("massimo", n1)
else:
    print("massimo", n2)

# Oppure con le funzioni apposite
print("minimo", min(n1, n2))
```

```
print("valore assoluto della differenza", abs(n1 - n2))
```

Esercizio 1-2

Progettare il programma `oddNumber.py` che:

- chieda all'utente di fornire un numero dispari (ovviamente intero...)
- se il numero fornito è pari, chieda di nuovo all'utente di fornire un numero dispari, dove aver visualizzato una segnalazione d'errore
- se il numero fornito (al primo o secondo tentativo) è dispari, scriva un messaggio di congratulazioni all'utente, riportando anche il numero fornito (es. "Bravo! Il numero 25 è dispari")
- se, invece, neanche il secondo numero fornito è dispari, scriva un messaggio di disappunto. Attenzione: se il primo numero fornito è dispari, il programma NON deve chiedere un secondo numero.

Soluzione

Nella soluzione sfruttiamo il fatto che un numero pari è per definizione divisibile per 2. Quindi il resto della divisione per 2 (operatore `%`) è sempre 0 per un numero pari

```
number = int(input("Inserires un numero dispari"))
if number % 2 == 0:
    number = input(int(("Errore. Inserires un numero dispari")))
if number % 2 == 0:
    print("Errore! Avevo chiesto un numero dispari!")
else:
    print("Bravo! Il numero", number, "è dispari")
```

Esercizio 1-3

Progettare il programma `isLeapYear.py` che segnali all'utente se il numero intero positivo che ha introdotto corrisponde a un anno bisestile oppure no.

- Ricordare che un anno è **bisestile** se è divisibile per 4. Fanno eccezione gli anni divisibili per 100, che non sono bisestili, e gli anni divisibili per 400, che invece sono bisestili (pur essendo divisibili anche per 100): tali eccezioni esistono però solo dopo l'adozione del calendario gregoriano, che avvenne nel **1582**.
- Il programma NON dovrà accettare il dato in ingresso (visualizzando, quindi, un messaggio d'errore) quando è un numero intero non positivo.

Il programma dovrà essere collaudato con i seguenti valori:

- un valore negativo (segnalazione d'errore)
- il valore zero (segnalazione d'errore)
- l'anno 1582 (che non è bisestile, non essendo divisibile per 4)
- anni maggiori di 1582
- un anno divisibile per 400, quindi bisestile (es. 2000)
- un anno divisibile per 100 ma non per 400, quindi non bisestile (es. 1900)
- un anno divisibile per 4 ma non per 100, quindi bisestile (es. 2016)
- un anno non divisibile per 4, quindi non bisestile (es. 2017)
- anni minori di 1582
- un anno divisibile per 400, quindi bisestile (es. 1200)
- un anno divisibile per 100 ma non per 400, comunque bisestile (es. 1400)
- un anno divisibile per 4 ma non per 100, quindi bisestile (es. 1580)
- un anno non divisibile per 4, quindi non bisestile (es. 1581)

Risolvere l'esercizio in due modi:

- senza utilizzare gli operatori booleani (and, or e not)
- utilizzando gli operatori booleani

Prima di scrivere codice, ragioniamo con i diagrammi di flusso...

Soluzione

```
year = int(input("Inserire un anno: "))
if year <= 0:
    print("Errore. Inserire solo anni positivi")
elif year >= 1582:
    if year % 400 == 0:
        print("Il", year, "è bisestile")
    elif year % 100 == 0:
        print("Il", year, "NON è bisestile")
    elif year % 4 == 0:
        print("Il", year, "è bisestile")
elif year % 4 == 0:
    print("Il", year, "è bisestile")
else:
    print("Il", year, "NON è bisestile")
```

Esercizio 2-4

Progettare il programma `checkCircles.py` che:

- chieda all'utente di inserire la descrizione di due circonferenze nel piano cartesiano:
 - coordinata x del centro della prima circonferenza
 - coordinata y del centro della prima circonferenza
 - raggio della prima circonferenza
 - coordinata x del centro della seconda circonferenza
 - coordinata y del centro della seconda circonferenza
 - raggio della seconda circonferenza
- visualizzi uno dei messaggi seguenti:
 - Le due circonferenze sono coincidenti
 - Le due circonferenze sono tangenti esternamente
 - Le due circonferenze sono tangenti internamente
 - Le due circonferenze sono secanti
 - Le due circonferenze sono concentriche (ma non coincidenti)
 - Le due circonferenze sono una interna all'altra, ma non concentriche
 - Le due circonferenze sono (reciprocamente) esterne

Per dubbi 'matematici': <http://www.ripmat.it/mate/f/fj/fje.html>

Soluzione

```
x1 = float(input("Circonferenza 1 - Centro x: "))
y1 = float(input("Circonferenza 1 - Centro y: "))
r1 = float(input("Circonferenza 1 - Raggio: "))
x2 = float(input("Circonferenza 2 - Centro x: "))
y2 = float(input("Circonferenza 2 - Centro y: "))
r2 = float(input("Circonferenza 2 - Raggio: "))
if r1 <= 0 or r2 <= 0 :
    print("Errore: entrambi i raggi devono essere positivi")
else :
    # calcolo la distanza tra i centri (teorema di Pitagora...)
    d = ((x1 - x2)**2 + (y1 - y2)**2)**(1/2)
    # calcolo il valore assoluto della differenza tra i raggi
    diff = r1 - r2
    if diff < 0 :
        diff = -diff
    # calcolo la somma dei raggi
    sum = r1 + r2
```

```

# alternative multiple per decidere...
if d == 0 :
    if r1 == r2 : # qui è vero che d == 0 and r1 == r2
        print("Le due circonferenze sono coincidenti")
    else :
        # qui è vero che d == 0 and r1 != r2
        print("Le due circonferenze sono concentriche (ma non coincidenti)")
elif 0 < d < diff :
    print("Le due circonferenze sono una interna all'altra, ma non concentriche")
elif d == diff :
    print("Le due circonferenze sono tangenti internamente")
elif diff < d < sum :
    print("Le due circonferenze sono secanti")
elif d == sum :
    print("Le due circonferenze sono tangenti esternamente")
else : # qui è vero che d > sum, non c'è bisogno di verificarlo
    print("Le due circonferenze sono (reciprocamente) esterne")

```

Exercizio 2-5

Progettare il programma `romanNumeral.py` che converta un numero intero positivo nel corrispondente valore espresso nel sistema di numerazione romano.

I numeri romani sono costituiti da sequenze di sei diverse possibili cifre, aventi il valore seguente:

- I = 1
- V = 5
- X = 10
- L = 50
- C = 100
- D = 500
- M = 1000

Queste sono le regole di conversione:

- si possono rappresentare soltanto i numeri minori di 4000
- (come nel sistema di numerazione decimale posizionale) le migliaia, le centinaia, le decine e le unità vengono espresse separatamente e rappresentate in questo ordine
- i numeri da 1 a 9 sono rappresentati, in ordine crescente, da I, II, III, IV, V, VI, VII, VIII, IX; come si può notare, una lettera I che precede la lettera V o X rappresenta un'unità che viene sottratta dal valore del numero; inoltre, non si possono avere più di tre I consecutive

- le decine sono gestite come le unità, tranne per il fatto che, al posto delle lettere I, V e X, si usano rispettivamente le lettere X, L e C
- le centinaia sono gestite come le unità, tranne per il fatto che, al posto delle lettere I, V e X, si usano rispettivamente le lettere C, D e M
- Il programma deve chiedere all'utente un numero intero positivo minore di 4000 e visualizzare il corrispondente numero romano. Esempio: 1978 diventa MCMLXXVIII.

Soluzione

```
n = int(input("Numero intero da convertire in numero romano (da 1 a 3999) "))
if n > 0 and n < 4000 :
    roman = ""
    thou = n // 1000
    if thou == 1 :
        roman = roman + "M"
    elif thou == 2 :
        roman = roman + "MM"
    elif thou == 3 :
        roman = roman + "MMM"
    hund = (n - thou * 1000) // 100
    if hund == 1 :
        roman = roman + "C"
    elif hund == 2 :
        roman = roman + "CC"
    elif hund == 3 :
        roman = roman + "CCC"
    elif hund == 4 :
        roman = roman + "CD"
    elif hund == 5 :
        roman = roman + "D"
    elif hund == 6 :
        roman = roman + "DC"
    elif hund == 7 :
        roman = roman + "DCC"
    elif hund == 8 :
        roman = roman + "DCCC"
    elif hund == 9 :
        roman = roman + "CM"
    tens = (n - thou * 1000 - hund * 100) // 10
    if tens == 1 :
        roman = roman + "X"
```

```

elif tens == 2 :
    roman = roman + "XX"
elif tens == 3 :
    roman = roman + "XXX"
elif tens == 4 :
    roman = roman + "XL"
elif tens == 5 :
    roman = roman + "L"
elif tens == 6 :
    roman = roman + "LX"
elif tens == 7 :
    roman = roman + "LXX"
elif tens == 8 :
    roman = roman + "LXXX"
elif tens == 9 :
    roman = roman + "XC"
units = n - thou * 1000 - hund * 100 - tens * 10
if units == 1 :
    roman = roman + "I"
elif units == 2 :
    roman = roman + "II"
elif units == 3 :
    roman = roman + "III"
elif units == 4 :
    roman = roman + "IV"
elif units == 5 :
    roman = roman + "V"
elif units == 6 :
    roman = roman + "VI"
elif units == 7 :
    roman = roman + "VII"
elif units == 8 :
    roman = roman + "VIII"
elif units == 9 :
    roman = roman + "IX"
print(roman)
else :
    print("Numero errato")

```