

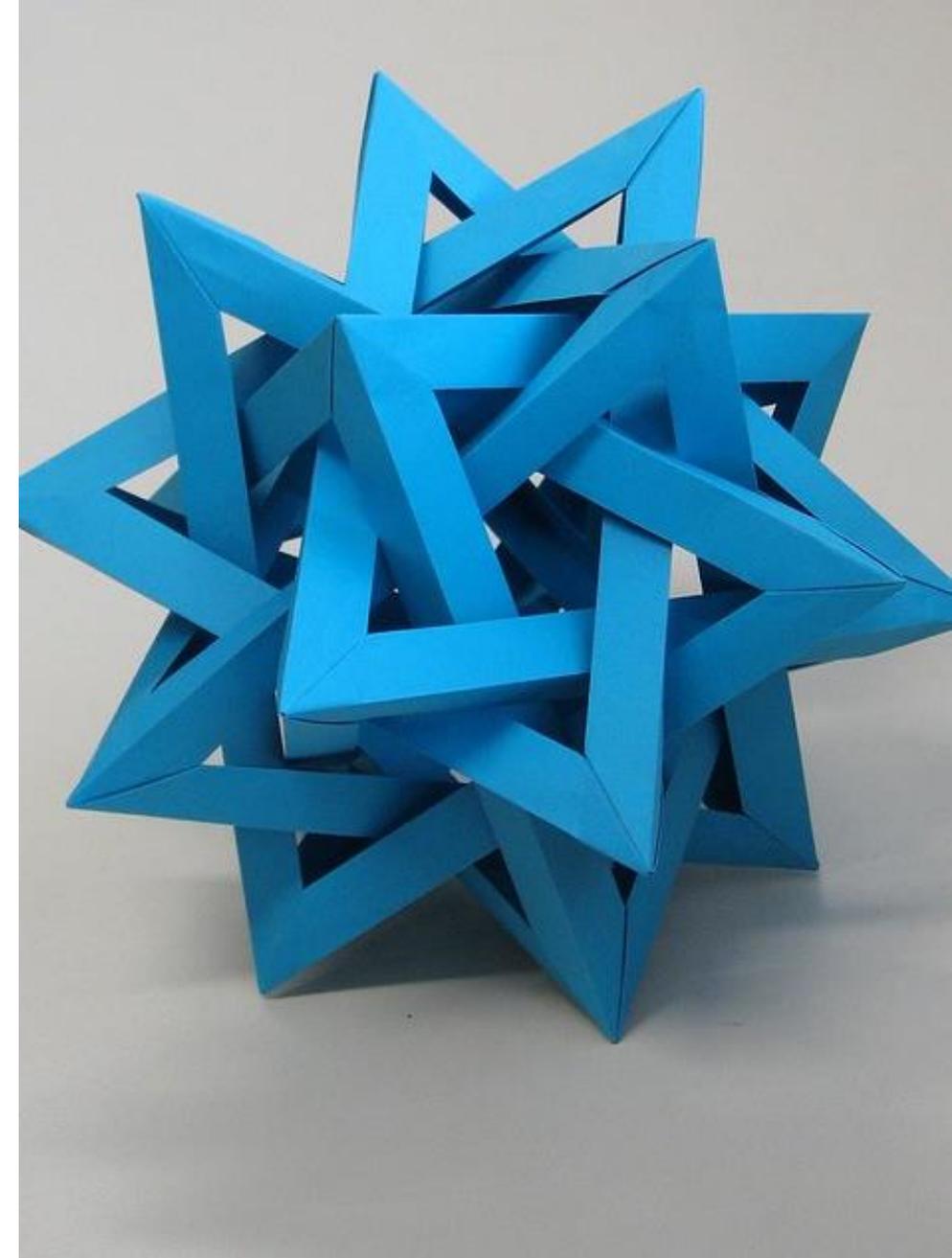


Politecnico
di Torino

Dipartimento
di Automatica e Informatica

Laboratorio 3

SCELTE LOGICHE ED ESPRESSIONI BOOLEANE



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

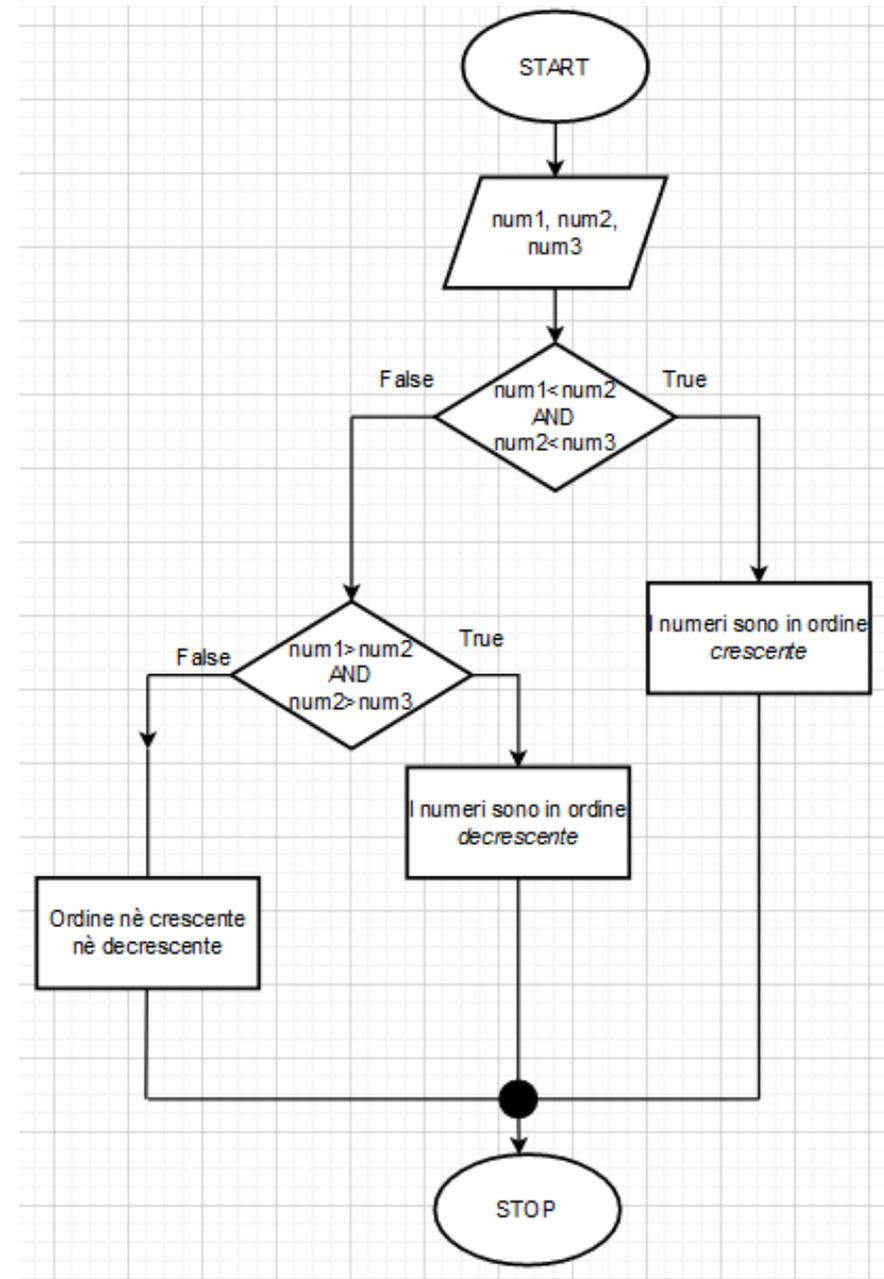
Esercizio 1

Esercizio 1. Scrivete un programma che legga tre numeri e visualizzi il messaggio “increasing” se sono in ordine crescente, “decreasing” se sono in ordine decrescente e “neither” se non sono né in ordine crescente né in ordine decrescente. In questo esercizio crescente significa strettamente crescente, cioè ciascun valore deve essere maggiore del precedente (analogo significato ha il termine decrescente): la sequenza 3 4 4, quindi, non va considerata crescente. [P3.5]

Esercizio 1 – *il flowchart*

Istruzione **elif**

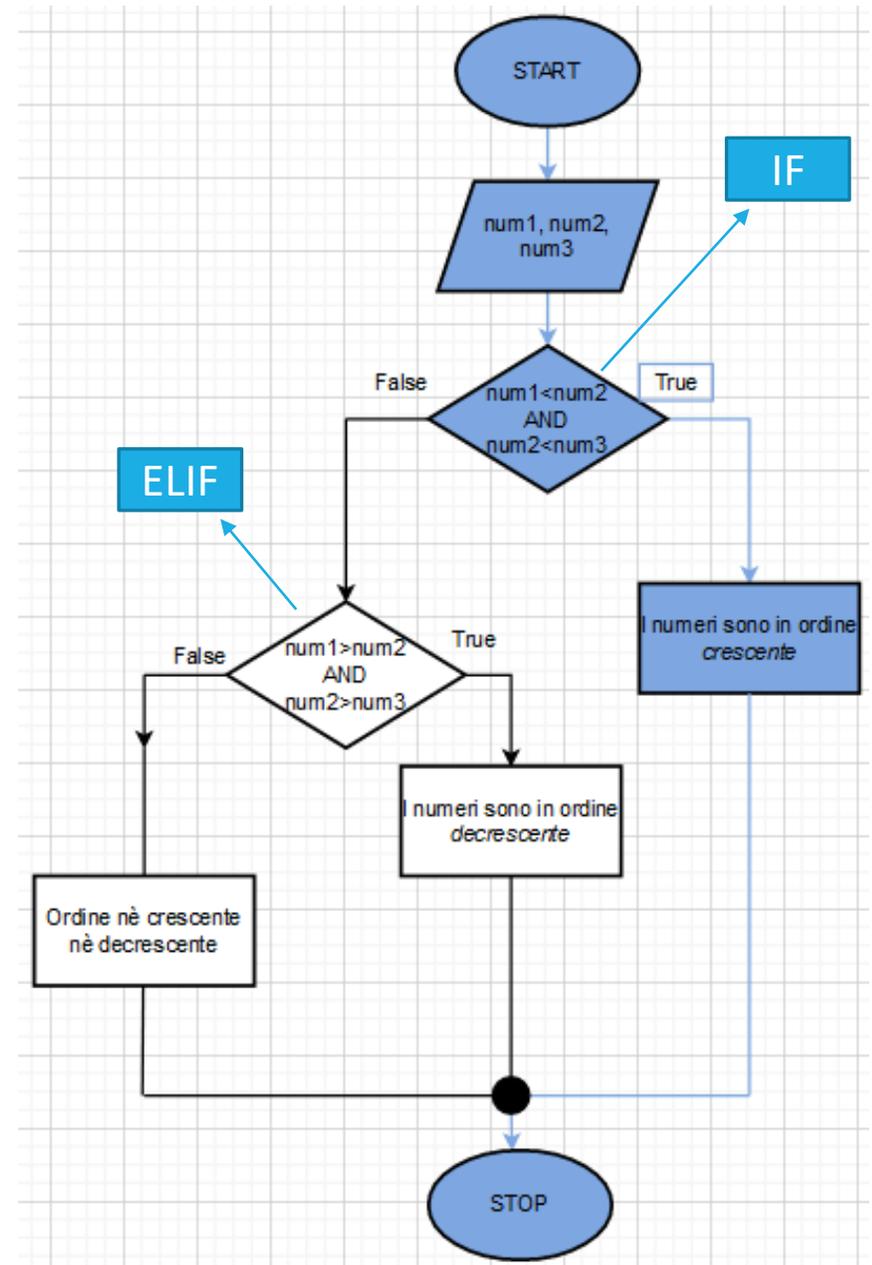
- Abbreviazione di: *else, if...*
- Appena una delle condizioni di test è vera, il relativo blocco di istruzioni viene eseguito
 - I rimanenti test non vengono valutati
- Se nessun test ha successo, viene eseguito l' *else* finale



Esercizio 1 – collaudare le diramazioni

Caso con 3 numeri crescenti:

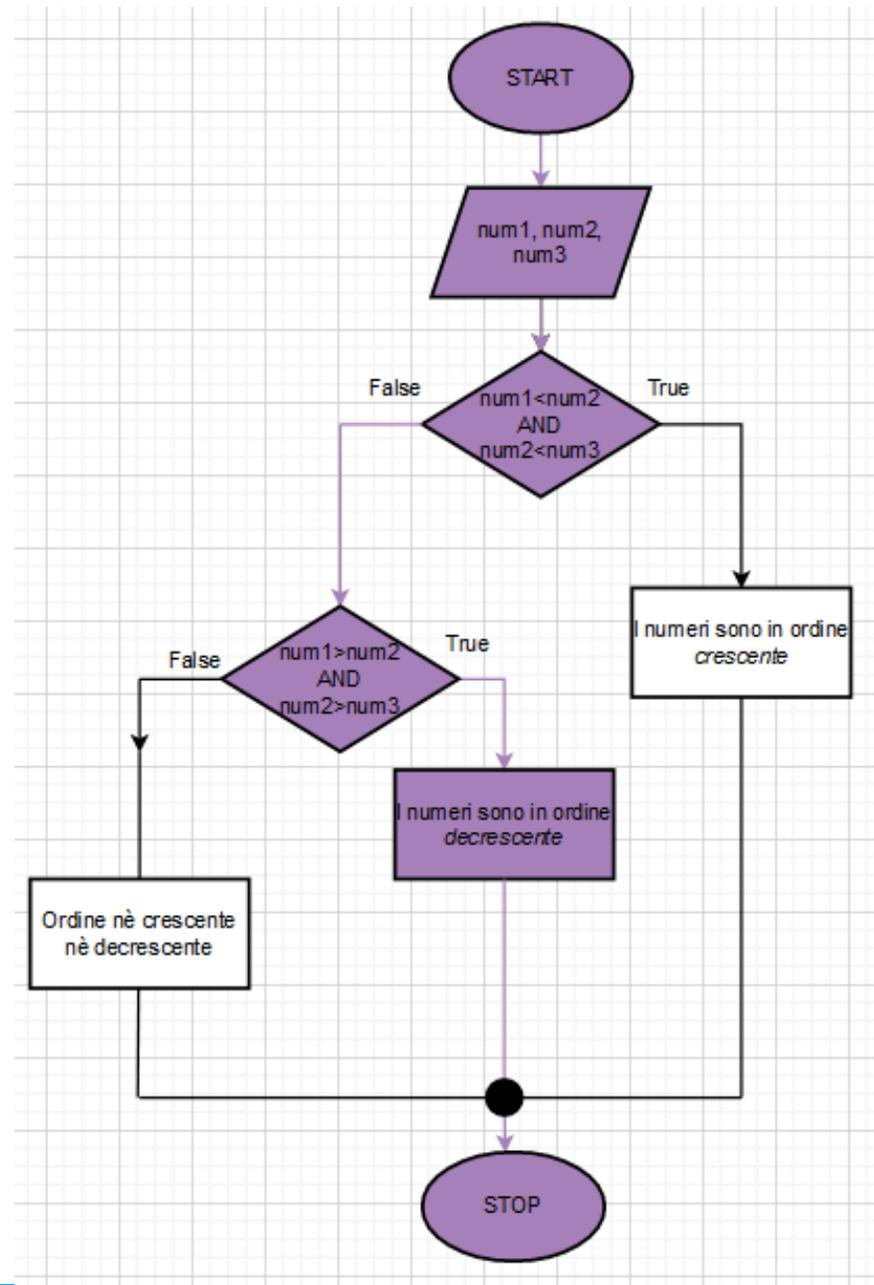
num1 = 5
num2 = 8
num3 = 12



Esercizio 1 – collaudare le diramazioni

Caso con 3 numeri decrescenti:

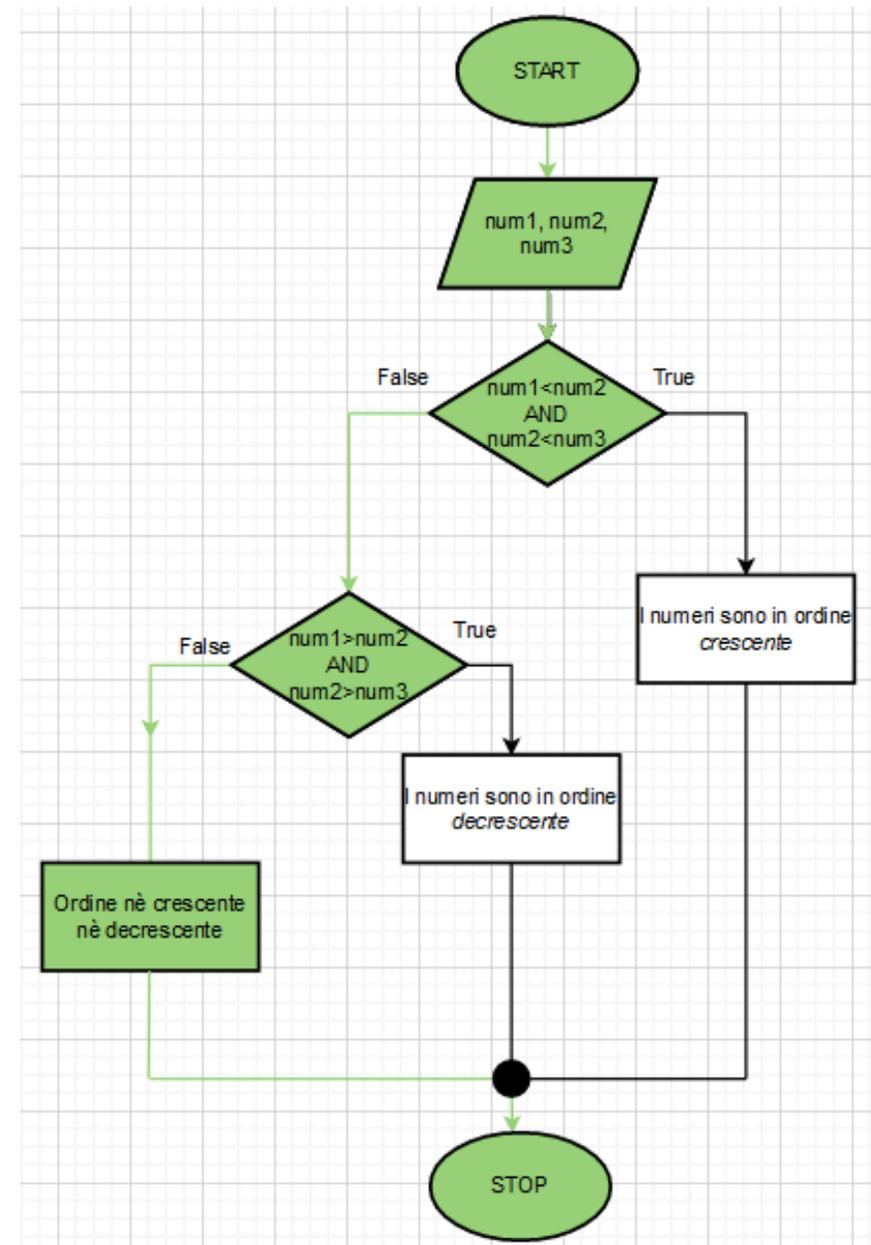
num1 = 20
num2 = 14
num3 = 2



Esercizio 1 – collaudare le diramazioni

Caso con 3 numeri in nessun ordine:

num1 = 30
num2 = 25
num3 = 45

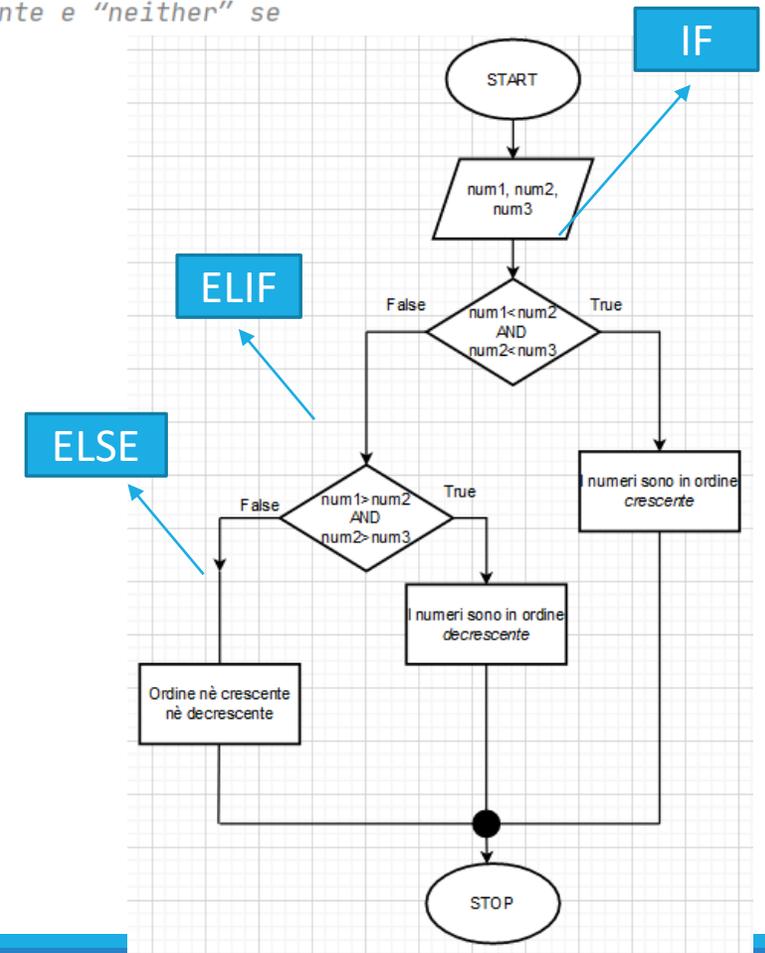


Esercizio 1 – *il codice Python*

```
## LAB3_es1
# Scrivete un programma che legga tre numeri e visualizzi il messaggio "increasing" se
# sono in ordine crescente, "decreasing" se sono in ordine decrescente e "neither" se
# non sono né in ordine crescente né in ordine decrescente.

# Chiedere all'utente di inserire i tre numeri
num1 = float(input("Inserire il primo numero: "))
num2 = float(input("Inserire il secondo numero: "))
num3 = float(input("Inserire il terzo numero: "))

# Determinare l'ordine dei numeri e mostrarne lo stato
if num1 < num2 and num2 < num3 :
    print("Sono in ordine crescente.")
elif num1 > num2 and num2 > num3 :
    print("Sono in ordine decrescente.")
else :
    print("Non sono né in ordine crescente né decrescente.")
```



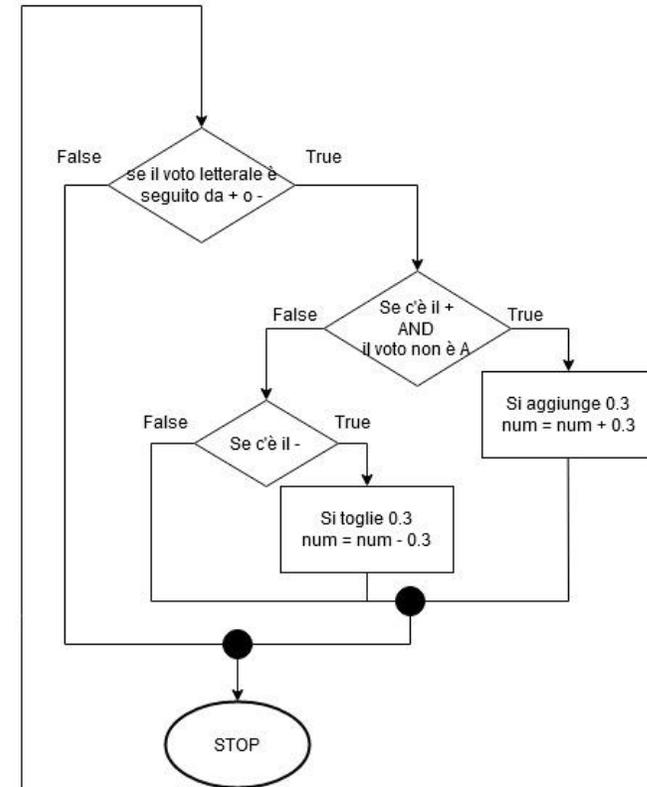
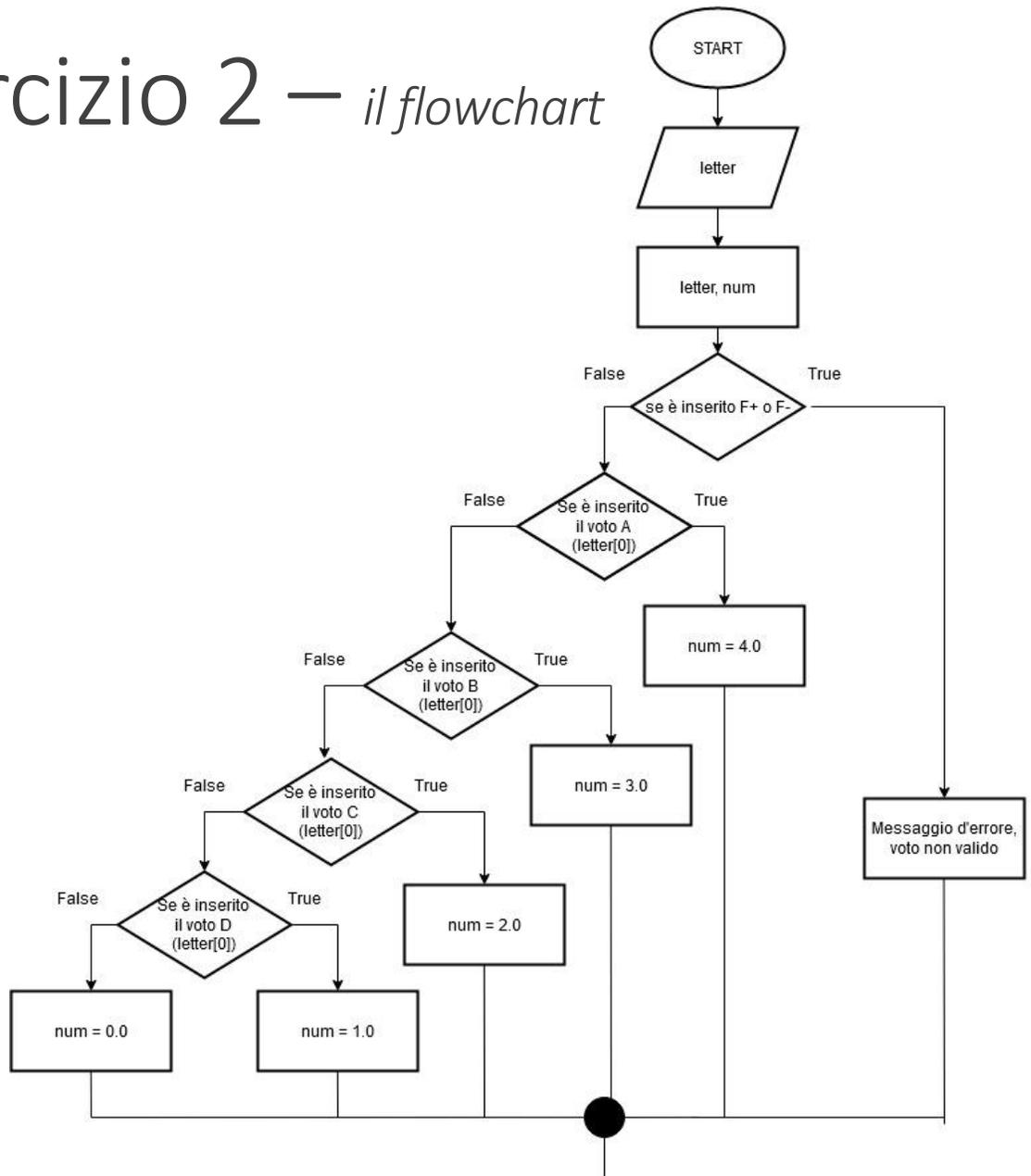
Esercizio 2

Esercizio 2. Scrivete un programma che traduca un voto in lettere nel corrispondente voto numerico. I voti in lettere sono A, B, C, D e F, eventualmente seguiti da un segno + o -. I loro valori numerici sono, nell'ordine, 4, 3, 2, 1 e 0. I voti F+ e F- non esistono. Un segno + aumenta il voto numerico di 0.3, mentre un segno - lo diminuisce della stessa quantità. Il voto A+ è comunque uguale a 4.0.

Enter a letter grade: **B-**

The numeric value is 2.7. [P3.12]

Esercizio 2 – il flowchart



NB:

Regola generale

- Non fidarsi MAI dell'input dell'utente (non assumere mai che sia corretto né sensato)
- Quando si legge un input dall'utente, **controllare sempre che contenga valori accettabili**, prima di continuare l'esecuzione
- Se il valore letto non è accettabile, stampare un messaggio e :
 - Richiedere di inserire nuovamente il valore (vedi i Cicli, Capitolo 4)
 - Uscire del programma:

```
from sys import exit
exit("Valore non accettabile")
```

Esercizio 2 — *il codice Python*

```
## LAB3_es2
from sys import exit

# Legge il voto in lettere inserito dall'utente
letter = input("Inserire un voto in lettere (es: A o A+ o A-): ").upper()

# Controlla validità input e converte la parte letterale in voto
if len(letter) > 1 and letter[0] == "F":
    exit("Errore: il voto interito non è valido.")
elif letter[0] == "A" :
    num = 4.0
elif letter[0] == "B" :
    num = 3.0
elif letter[0] == "C" :
    num = 2.0
elif letter[0] == "D" :
    num = 1.0
else :
    num = 0.0

# Tiene conto del + o del -, se presenti
if len(letter) > 1:
    if letter[1] == "+" and letter[0] != "A" :
        num = num + 0.3
    elif letter[1] == "-" :
        num = num - 0.3

# Mostra il risultato
print("Il valore numerico del voto letterale %s è %.2f." %(letter[0], num))
```

NB:

- Uso di .upper()
- Gestione dell'input dell'utente

Esercizio 3

- Esercizio 3. Scrivete un programma che legga una stringa e visualizzi i messaggi appropriati, dopo aver verificato se:
- a. contiene soltanto lettere
 - b. contiene soltanto lettere maiuscole
 - c. contiene soltanto lettere minuscole
 - d. contiene soltanto cifre
 - e. contiene soltanto lettere e cifre
 - f. inizia con una lettera maiuscola
 - g. termina con un punto [P3.17]

Esercizio 3 — *documentazione, metodi utili*

String Methods

Strings implement all of the [common](#) sequence operations, along with the additional methods described below.

Strings also support two styles of string formatting, one providing a large degree of flexibility and customization (see `str.format()`, [Format String Syntax](#) and [Custom String Formatting](#)) and the other based on C `printf` style formatting that handles a narrower range of types and is slightly harder to use correctly, but is often faster for the cases it can handle ([printf-style String Formatting](#)).

The [Text Processing Services](#) section of the standard library covers a number of other modules that provide various text related utilities (including regular expression support in the `re` module).



<https://docs.python.org/3/library/stdtypes.html?highlight=islower>

`str.isupper()`

Return `True` if all cased characters [\[4\]](#) in the string are uppercase and there is at least one cased character, `False` otherwise.

```
>>> 'BANANA'.isupper()
True
>>> 'banana'.isupper()
False
>>> 'baNana'.isupper()
False
>>> ''.isupper()
False
```

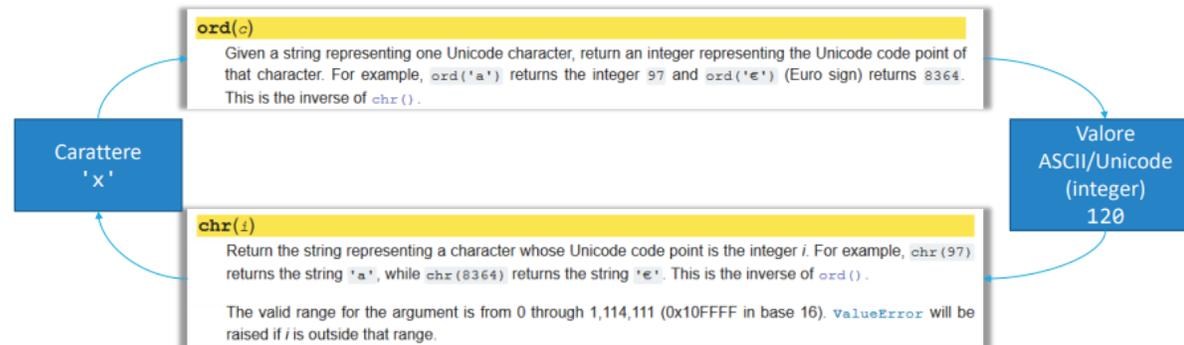
Esercizio 3 — caratteri Unicode

Caratteri

- I caratteri sono memorizzati come valori interi
 - Vedere il subset ASCII subset nella tabella [Unicode](#), Appendice A
 - Per esempio, la lettera 'H' ha il valore ASCII 72
- Python usa i caratteri **Unicode**
 - Unicode definire oltre 100,000 caratteri
 - Unicode è stato creato capace di codificare il testo sostanzialmente in tutte le lingue scritte
 - <https://home.unicode.org/> e <http://www.unicode.org/charts/>



Conversione dei caratteri



Esercizio 3 — *il codice Python*

```
## LAB3_es3

# Legge la stringa in input
string = input("Inserire una stringa: ")

# Controlla ogni proprietà e, se presente, visualizza messaggio
if string.isalpha() :
    print("La stringa contiene solo lettere.")

if string.isupper() :
    print("Tutte le lettere della stringa sono MAIUSCOLE.")

if string.islower() :
    print("Tutte le lettere della stringa sono minuscole.")

if string.isdigit() :
    print("La stringa contiene solo numeri.")

if string.isalnum() :
    print("La stringa contiene solo lettere e numeri.")

if string[0] >= "A" and string[0] <= "Z" :
    print("La stringa inizia con la lettera maiuscola.")

if string.endswith(".") :
    print("La stringa finisce con un punto.")
```

NB:

- Nel confronto fra caratteri sono comparati i loro valori ordinali (Unicode)

Esercizio 4

Esercizio 4. L'algoritmo seguente individua la stagione (Spring, Summer, Fall o Winter, cioè, rispettivamente, primavera, estate, autunno o inverno) a cui appartiene una data, fornita come mese e giorno.

Se mese è 1, 2 o 3, stagione = "Winter"

Altrimenti se mese è 4, 5 o 6, stagione = "Spring"

Altrimenti se mese è 7, 8 o 9, stagione = "Summer"

Altrimenti se mese è 10, 11 o 12, stagione = "Fall"

Se mese è divisibile per 3 e giorno ≥ 21

Se stagione è "Winter", stagione = "Spring"

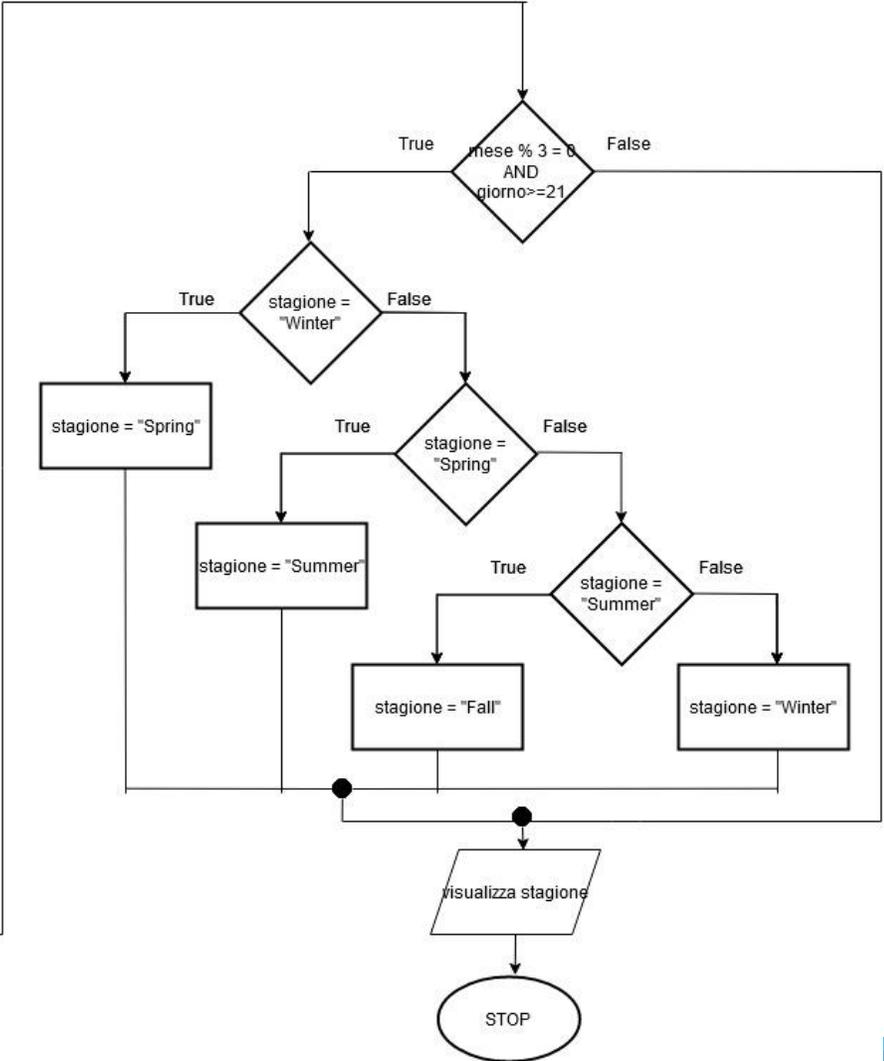
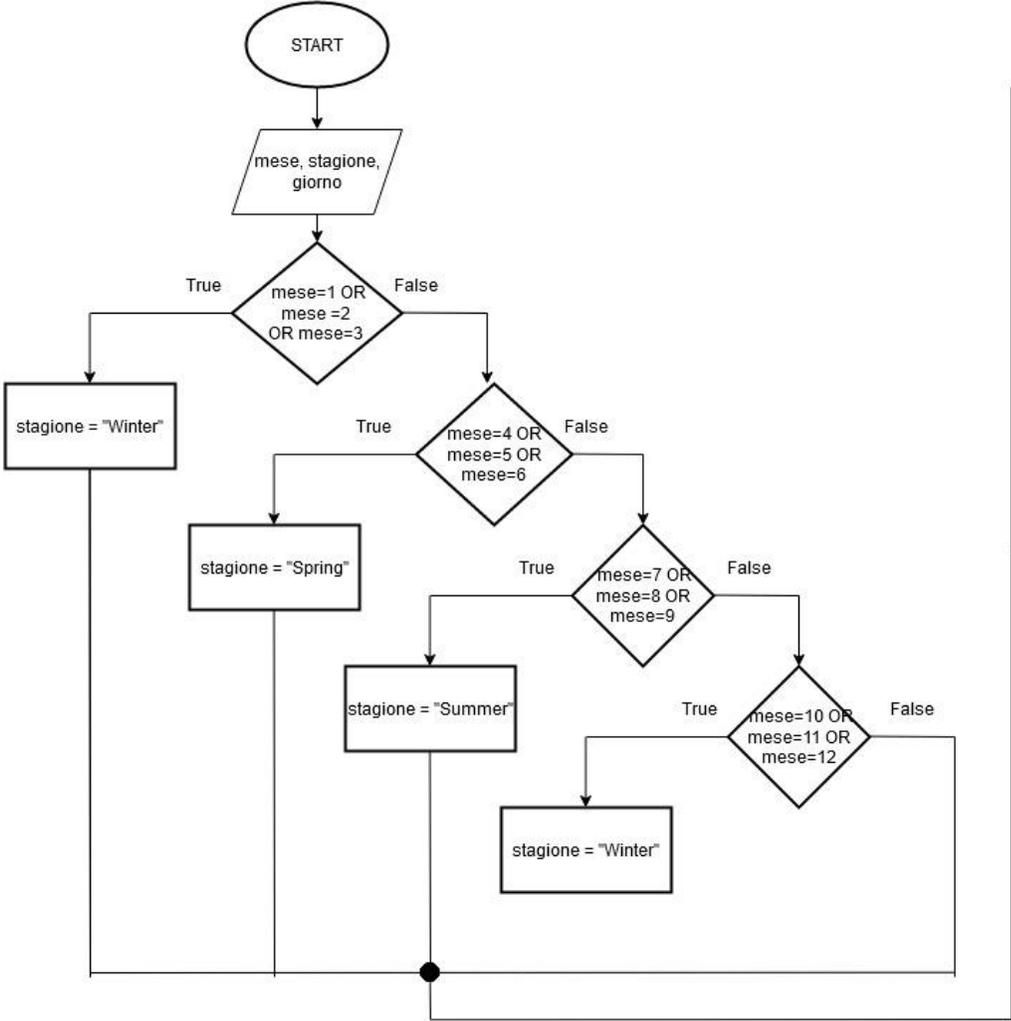
Altrimenti se stagione è "Spring", stagione = "Summer"

Altrimenti se stagione è "Summer", stagione = "Fall"

Altrimenti stagione = "Winter"

Scrivete un programma che chieda all'utente un mese e un giorno e, poi, visualizzi la stagione determinata da questo algoritmo. [P3.20]

Esercizio 4 – flowchart



Esercizio 4 — *il codice Python*

```
## LAB3_es4

# Chiede all'utente di inserire un mese e un giorno
month = int(input("Inserire il mese: "))
day = int(input("Inserire il giorno: "))

# Determina la stagione
if month >= 1 and month <= 3 :
    season = "Inverno"
elif month >= 4 and month <= 6 :
    season = "Primavera"
elif month >= 7 and month <= 9 :
    season = "Estate"
elif month >= 10 and month <= 12 :
    season = "Autunno"

if month % 3 == 0 and day >= 21 :
    if season == "Inverno" :
        season = "Primavera"
    elif season == "Primavera" :
        season = "Estate"
    elif season == "Estate" :
        season = "Autunno"
    else :
        season = "Inverno"

# Visualizza la stagione
print("Il giorno %d/%d è in %s" %(day, month, season))
```

NB:

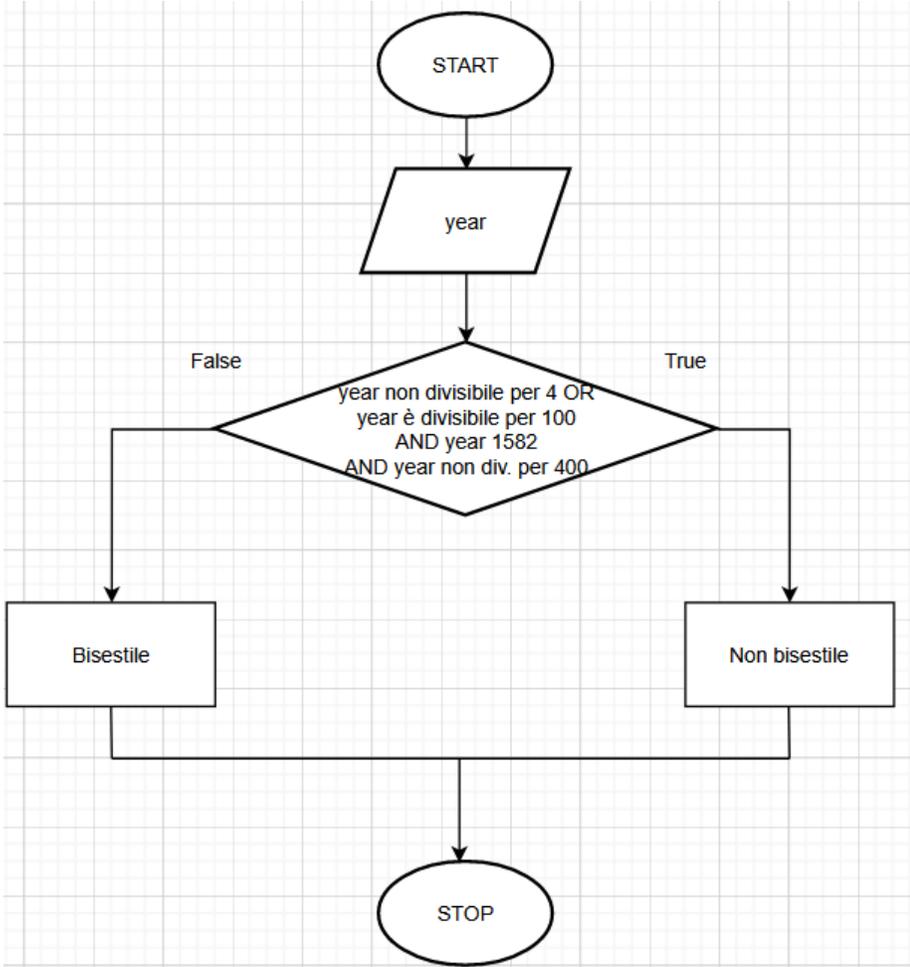
- Conversione in intero dell'input dell'utente
- Uso degli AND al posto degli OR dello pseudocodice, è equivalente
- Formattazione del `print()` finale

Esercizio 5

Esercizio 5. Un anno di 366 giorni viene detto bisestile (leap) e serve a mantenere il calendario sincronizzato con il Sole, dal momento che la Terra vi ruota attorno una volta ogni 365.25 giorni. In realtà, questo numero non è esatto e per tutte le date successive al 1582 si applica la correzione gregoriana: solitamente gli anni divisibili per 4, come il 1996, sono bisestili, ma gli anni divisibili per 100, come il 1900, non lo sono; come eccezione all'eccezione, gli anni divisibili per 400, come il 2000, sono bisestili. Scrivete un programma che chieda all'utente un anno e determini se si tratta di un anno bisestile, usando un unico enunciato if (con gli opportuni operatori booleani).

[P3.27]

Esercizio 5 – *flowchart*



Esercizio 5 — *il codice Python*

```
## LAB3_es5

# Chiede all'utente di inserire l'anno
year = int(input("Inserire un anno (es.1990): "))

# Determina se l'anno è bisestile
if year % 4 != 0 or year % 100 == 0 and year > 1582 and year % 400 != 0 :
    print(year, "non è bisestile.")
    # oppure
    # print("L'anno %d non è bisestile." %year)
else :
    print(year, "è bisestile.")
    # oppure
    # print("L'anno %d è bisestile." %year)
```

Esercizio 6

Esercizio 6. Considerando i valori numerici dei voti spiegati nell'esercizio 2, scrivete un programma che traduca un numero compreso tra 0 e 4 nel voto letterale più vicino. Ad esempio, il numero 2.8 (che potrebbe essere la media di più voti) deve essere tradotto come B-. Risolvete i casi di parità in favore del voto migliore: ad esempio, 2.85 deve essere tradotto come B. [P3.13]

Esercizio 6 — *il codice Python*

```
## LAB3_es6
# riferimento all'es.2 di questo laboratorio

# Legge l'input dall'utente
num = float(input("Inserire un voto numerico compreso tra 0 e 4: "))

# Converte al voto letterale più vicino
if num >= 3.85 :
    letter = "A"
elif num >= 3.5 :
    letter = "A-"
elif num >= 3.15 :
    letter = "B+"
elif num >= 2.85 :
    letter = "B"
elif num >= 2.5 :
    letter = "B-"
elif num >= 2.15 :
    letter = "C+"
elif num >= 1.85 :
    letter = "C"
elif num >= 1.5 :
    letter = "C-"
elif num >= 1.15 :
    letter = "D+"
elif num >= 0.85 :
    letter = "D"
elif num >= 0.5 :
    letter = "D-"
else :
    letter = "F"

# Mostra il risultato
print("Il voto %.2f convertito in voto letterale è: %s." %(num ,letter))
```

NB:

- Importanza uso dell'`elif`. Cosa succederebbe usando degli `if`?

Esercizio 7

Esercizio 7. Scrivete un programma che calcoli le tasse secondo questo schema. [P3.25]

Per stato civile “non coniugato” e reddito imponibile superiore a	ma non superiore a	le tasse sono	della somma superiore a
\$ 0	\$ 8000	10%	\$ 0
\$ 8000	\$ 32 000	\$ 800 + 15%	\$ 8000
\$ 32 000		\$ 4400 + 25%	\$ 32 000

Per stato civile “coniugato” e reddito imponibile superiore a	ma non superiore a	le tasse sono	della somma superiore a
\$ 0	\$ 16 000	10%	\$ 0
\$ 16 000	\$ 64 000	\$ 1600 + 15%	\$ 16 000
\$ 64 000		\$ 8800 + 25%	\$ 64 000

Esercizio 7 — *il codice Python*

```
### LAB3_es7
#

# Legge input dell'utente
status = input("Sei sposato/a o single? (M or S)? ")
status = status.upper()

income = float(input("Qual è il tuo reddito? "))

# Calcola le tasse
if status == "S" :
    if income >= 0 and income <= 8000 :
        tax = income * 0.10
    elif income > 8000 and income <= 32000 :
        tax = 800 + (income - 8000) * 0.15
    elif income >= 32000 :
        tax = 4400 + (income - 32000) * 0.25
elif status == "M" :
    if income >= 0 and income <= 16000 :
        tax = income * 0.10
    elif income > 16000 and income <= 64000 :
        tax = 1600 + (income - 16000) * 0.15
    elif income >= 64000 :
        tax = 8800 + (income - 64000) * 0.25

# Mostra il risultato
print("Le tasse ammontano a: %.2f" %tax)
```

Esercizio 8

Esercizio 8. Scrivete un programma per la conversione di unità di misura che chieda all'utente da quale unità (scegliendo tra: ml, l, g, kg, mm, cm, m, km) e verso quale unità (scegliendo tra: fl. oz, gal, oz, lb, in, ft, mi) vuole effettuare una conversione, rifiutando conversioni incompatibili (come, ad esempio, gal \rightarrow km). Chiedete, poi, il valore da convertire e, infine, visualizzate il risultato:

Convert from? **gal**

Convert to? **ml**

Value? **2.5**

2.5 gal = 9463.5 ml

[P3.26]

Esercizio 8 — *il codice Python*

```
## LAB3_es8
#
from sys import exit

# Legge l'unità di misura da convertire da... a...
fromUnit = input("Convertire da? ")
toUnit = input("Convertire a? ")

# Unità di volume
if fromUnit == "fl. oz" and toUnit == "ml" :
    factor = 29.5735
elif fromUnit == "fl. oz" and toUnit == "l" :
    factor = 0.0295735
elif fromUnit == "gal" and toUnit == "ml" :
    factor = 3785.41
elif fromUnit == "gal" and toUnit == "l" :
    factor = 3.78541

# Unità di massa
elif fromUnit == "oz" and toUnit == "g" :
    factor = 28.3495
elif fromUnit == "oz" and toUnit == "kg" :
    factor = 0.0283495
elif fromUnit == "lb" and toUnit == "g" :
    factor = 453.592
elif fromUnit == "lb" and toUnit == "kg" :
    factor = 0.453592

# Unità di distanza
elif fromUnit == "in" and toUnit == "mm" :
    factor = 25.4
elif fromUnit == "in" and toUnit == "cm" :
    factor = 2.54
elif fromUnit == "in" and toUnit == "m" :
    factor = 0.0254
elif fromUnit == "in" and toUnit == "km" :
    factor = 0.0000254
elif fromUnit == "ft" and toUnit == "mm" :
    factor = 304.8
elif fromUnit == "ft" and toUnit == "cm" :
    factor = 30.48
elif fromUnit == "ft" and toUnit == "m" :
    factor = 0.3048
elif fromUnit == "ft" and toUnit == "km" :
    factor = 0.0003048
elif fromUnit == "mi" and toUnit == "mm" :
    factor = 1609344
elif fromUnit == "mi" and toUnit == "cm" :
    factor = 160934.4
elif fromUnit == "mi" and toUnit == "m" :
    factor = 1609.344
elif fromUnit == "mi" and toUnit == "km" :
    factor = 1.609344

# Unità di misura non compatibili
else :
    exit("Those units are not compatible.")

# Legge il valore da convertire
value = float(input("Valore? "))

# Calcola il risultato
result = value * factor

# Mostra il risultato
print(value, fromUnit, "=", result, toUnit)
```

Esercizio 9

Esercizio 9. Un supermercato premia i propri clienti con buoni spesa il cui importo dipende dalla quantità di denaro spesa in prodotti alimentari (*groceries*). Ad esempio, spendendo 50 dollari, si ottiene un buono spesa di importo pari all'otto per cento di quella somma. La tabella seguente mostra la percentuale usata per calcolare il buono spesa relativo a somme diverse. Scrivete un programma che calcoli e visualizzi il valore del buono spesa consegnato al cliente, sulla base della somma di denaro che ha speso nell'acquisto di prodotti alimentari. [P3.40]

Denaro speso	Percentuale del buono
Meno di \$ 10	Nessun buono
Da \$ 10 a \$ 60	8%
Da più di \$ 60 a \$ 150	10%
Da più di \$ 150 a \$ 210	12%
Più di \$ 210	14%

Esercizio 9 — *il codice Python*

```
## LAB3_es9
#
# Legge l'input dall'utente
cost = float(input("Inserire l'importo della spesa: "))

# Calcola l'importo del coupon
if cost < 10 :
    factor = 0
elif cost < 60 :
    factor = 8
elif cost < 150 :
    factor = 10
elif cost < 210 :
    factor = 12
else :
    factor = 14

coupon = cost * factor / 100

# Mostra il risultato
print("Hai ricevuto un coupon da $%.2f. (%d%% della tua spesa)" %
      (coupon, factor))
```