

esercizio

La sezione aurea è definita come $\phi = \frac{b}{a} = \frac{a+b}{b} = (1 + \sqrt{5})/2$ visualizzare il valore della sezione aurea

```
In [5]: import math

print("sezione aurea", (1 + math.sqrt(5)) / 2)

sezione aurea 1.618033988749895
```

esercizio

Scrivere la funzione primo che riceve come parametro un numero intero (sicuramente positivo) e restituisce True se questo è primo. Utilizzare la funzione per rispondere alla seguente domanda: quali dei seguenti numeri sono primi? 96553 15983567 17 (Risposta: il primo e il terzo)

La funzione deve avere una docstring significativa

```
In [2]: def primo(n: int) -> bool:
        ''' True se n è primo '''
        for d in range(2, n//2 + 1):
            if n % d == 0:
                return False
        return True

print(primo(15983567))

False
```

esercizio

Visualizzare la docstring della funzione precedentemente creata

```
In [8]: help(primo)

Help on function primo in module __main__:

primo(n: int) -> bool
    True se n è primo
```

esercizio

Scrivere la funzione volume che riceve come parametri le misure degli spigoli di un parallelepipedo rettangolo e restituisce il volume. Scrivere un programma che richiede in input le misure di un parallelepipedo rettangolo e, utilizzando la funzione volume stampa il volume.

```
In [21]: def volume(x: float, y: float, z: float) -> float:
        ''' volume parallelepipedo rettangolo '''
        return x * y * z
s1 = float(input('spigolo 1: '))
s2 = float(input('spigolo 2: '))
s3 = float(input('spigolo 3: '))
print('volume del parallelepipedo =', volume(s1,s2,s3))
```

```
spigolo 1: 2
spigolo 2: 3
spigolo 3: 4
volume del parallelepipedo = 24.0
```

In [22]: `help(volume)`

```
Help on function volume in module __main__:

volume(x: float, y: float, z: float) -> float
    volume parallelepipedo rettangolo
```

esercizio

Generare 10 numeri casuali nell'intervallo [10,100] e, per ognuno di questi, visualizzare tutti i suoi divisori primi

The randint(start, stop) includes both start and stop numbers while generating random integer
iniziare con `from random import randint`

```
In [27]: from random import randint
def primo(n: int) -> bool:
    ''' True se n è primo '''
    if n <= 0:
        return False          # non accettabile
    div = 2                    # primo divisore
    while div <= n / 2:
        if n % div == 0:
            return False
        div += 1
    return True

c = 0                          # numero valori generati
while c <= 10:
    v = randint(10,100)        # valore casuale
    print('valore generato:',v)
    d = 2                      # possibile divisore di v
    while d <= n/2:
        if v % d == 0 and primo(d):
            print(d,'è divisore primo di',v)
        d += 1
    c += 1
```

```
valore generato: 10
2 è divisore primo di 10
5 è divisore primo di 10
valore generato: 49
7 è divisore primo di 49
valore generato: 98
2 è divisore primo di 98
7 è divisore primo di 98
valore generato: 39
3 è divisore primo di 39
valore generato: 62
2 è divisore primo di 62
valore generato: 29
valore generato: 40
2 è divisore primo di 40
5 è divisore primo di 40
valore generato: 24
2 è divisore primo di 24
3 è divisore primo di 24
valore generato: 74
2 è divisore primo di 74
valore generato: 71
```

valore generato: 99
3 è divisore primo di 99

esercizio

funzione che trova tutti i fattori primi di un numero n

- parametro: n
- risultato: lista, contenente i fattori primi di n

algoritmo: scorrere tutti i valori d'interesse, e cercare i divisori

- x è divisore di n sse $n \% x == 0$

non considerare i fattori non primi

provare la funzione con valori inseriti dall'utente quando si trova un divisore x, dividere ripetutamente n per x, finché resta divisibile

```
In [4]: def fattori_primi(n: int) -> list:
        ''' restituisce una lista con i fattori primi di n '''
        fp = []          # fattori primi di n
        for d in range(2,n//2+1):
            if n % d == 0 and primo(d):
                fp.append(d)
        return fp

        v = int(input('valore intero positivo: '))
        print('fattori primi di',v,fattori_primi(v))
```

valore intero positivo: 150
fattori primi di 150 [2, 3, 5]