

---

# Esercizi: decisioni e iterazioni (2)

---

---

## Esercizio (senza iterazioni e selezioni)

- Realizzare una classe Square che oltre al costruttore fornisca i metodi getArea, getPerimeter e getDiagonal.
  - Scrivere un programma che prenda in input da tastiera un intero che rappresenta la lunghezza del lato di un quadrato e stampi:
    - L'area e il perimetro del quadrato
    - La lunghezza della diagonale del quadrato (applicare il teorema di Pitagora)
-

---

## Esercizio (senza iterazioni e selezioni)

- Realizzare una classe DigitExtractor che oltre al costruttore fornisce un metodo per estrarre la prossima cifra (in ordine inverso) di un numero
  - Scrivere un programma che legge da tastiera un numero intero di 5 cifre e stampi in ordine inverso le singole cifre che lo compongono
    - Es. se l'input è 42853 vengono stampate le cifre  
3  
5  
8  
2  
4
-

---

## Esercizio (senza cicli)

- Scrivere un programma che prende in input da tastiera 3 numeri in virgola mobile per poi stamparli in ordine crescente.



---

## Esercizio (senza cicli)

- Scrivere un programma che stampi
    - “OK” se legge “S” ,”SI”, “OK”, “certo” oppure “perche` no?”.
    - “Fine” se legge “N” o “No”
    - “Dato non corretto” altrimenti
  - A tal fine scrivere una classe InputChecker
-

---

## Esercizio (senza cicli)

- Scrivere un programma che prende in input da tastiera un numero fissato di parole (per esempio 4) e stampa la prima e l'ultima parola secondo l'ordine lessicografico
  - Come sopra ignorando la differenza tra maiuscole e minuscole
-

---

# Esercizio

Scrivere un programma che chieda all'utente di inserire il tasso di cambio fra dollari ed euro. Il programma legge alcuni valori in dollari e li converte in euro fino a che l'utente non digita "basta"

---

---

# Esercizio

- Scrivere un programma che legge in input un intero **n** e stampa un triangolo isoscele con **n** righe formate da `\*`.
- Esempio: se l'input è 3 stampa

```
  *  
 * * *  
* * * * *
```



---

## Esercizio

- Realizzare una classe **FibonacciGenerator** con metodo **nextNumber()** che genera il prossimo numero di Fibonacci
    - Si ricorda che  $f_0 = 1$ ,  $f_1 = 1$  e  $f_n = f_{n-1} + f_{n-2}$
  - Scrivere e testare un programma che dato in input un intero **n** stampa l'ennesimo numero di Fibonacci
-