

# Programmazione

Sessione di laboratorio

*Lab05 - Istruzioni if e if-else.*

# *Lone sum*

## *Esercizio*

Scrivere un metodo chiamato `loneSum` che, dati in input tre numeri positivi `a`, `b` e `c` di tipo `int`, restituisca la somma dei numeri che sono diversi fra di loro.

## *Esempi*

- `a = 1, b = 2 e c = 3: 6`
- `a = 3, b = 2 e c = 3: 2`
- `a = 3, b = 3 e c = 3: 0`

# *No teen sum*

## *Esercizio*

Scrivere un metodo chiamato `noTeenSum` che, dati in input tre numeri positivi  $a$ ,  $b$  e  $c$  di tipo `int`, restituisca la somma fra i numeri che non sono numeri *teen*: un numero è *teen* se è incluso nell'intervallo  $[13, 19]$  ed è diverso da 15 e da 16. Per evitare la duplicazione di codice, scrivere un metodo `fixTeen` che, dato in input un numero positivo  $a$  di tipo `int`, restituisca lo stesso numero se non è *teen*, altrimenti restituisca 0. Utilizzare poi il metodo `fixTeen` all'interno del metodo `noTeenSum`.

## *Esempi*

- $a = 1, b = 2$  e  $c = 3$ : 6
- $a = 2, b = 13$  e  $c = 1$ : 3
- $a = 2, b = 1$  e  $c = 14$ : 3
- $a = 2, b = 1$  e  $c = 15$ : 18

# *Round sum*

## *Esercizio*

Scrivere un metodo chiamato `roundSum` che, dati in input tre numeri positivi `a`, `b` e `c` di tipo `int`, restituisca la somma dei tre numeri arrotondati al multiplo di 10 più vicino: l'arrotondamento di un numero positivo è per difetto se l'ultima cifra è minore di 5, per eccesso altrimenti. Per evitare la duplicazione di codice, scrivere un metodo `round` che, dato in input un numero positivo `a` di tipo `int`, restituisca il suo arrotondamento. Utilizzare poi il metodo `round` all'interno del metodo `roundSum`.

## *Esempi*

- `a = 16, b = 17 e c = 18: 60`
- `a = 12, b = 13 e c = 14: 30`
- `a = 6, b = 4 e c = 4: 10`
- `a = 4, b = 4 e c = 4: 0`

## *Evenly spaced*

### *Esercizio*

Scrivere un metodo chiamato `evenlySpaced` che, dati in input tre numeri positivi `a`, `b` e `c` di tipo `int`, restituisca `true` se e solo se la differenza tra il valore di mezzo e quello minimo sia uguale alla differenza tra il valore massimo e quello di mezzo. Per semplificare la scrittura del codice, scrivere un metodo `min` (rispettivamente, `mid` e `max`) che, dati in input tre numeri positivi `a`, `b` e `c` di tipo `int`, restituisca il valore minimo (rispettivamente, di mezzo e massimo). Utilizzare poi i metodi `min`, `mid` e `max` all'interno del metodo `evenlySpaced`.

### *Esempi*

- `a = 2, b = 4 e c = 6: true`
- `a = 4, b = 6 e c = 2: true`
- `a = 4, b = 6 e c = 3: false`