

Esercizi

- Operatori
- Decisioni
- Loop
- Stringhe e array
- ...
- Progetto di riferimento
 - <https://github.com/egalli64/jse> (*modulo mex*)

Operatori

- 1 Distanza e tempo \rightarrow velocità media
- 2 Due punti (x_0, y_0) e (x_1, y_1) in un piano \rightarrow la loro distanza
 - Operatore radice quadrata?
- 3 Alesaggio e corsa in mm, numero cilindri \rightarrow cilindrata in cm cubi
- 4 Una temperatura in gradi Celsius \rightarrow la sua conversione in Fahrenheit
 - nove quinti di C più 32
- 5 (x, y) in un piano cartesiano \rightarrow punteggio in base alla distanza dal centro
 - Minore di 1 \rightarrow 10, in $[1, 5]$ \rightarrow 5, in $[5, 10]$ \rightarrow 1, maggiore di 10 \rightarrow 0

Decisioni

- 1 Un numero → “positive”, “negative”, o “zero”
- 2 Un intero → true se dispari
- 3 Un intero → “zero”, “one” ... “nine” se [0..9], altrimenti “other”
- 4 Il giorno della settimana, un flag che indica se siamo in vacanza → true se la sveglia deve suonare
- 5 Un numero in [0..100] → il corrispondente voto in ['A'..'F']
 - F ≤ 50, E in (50, 60], D in (60, 70], C in (70, 80], B in (80, 90], A > 90
- 6 Un intero → true se anno bisestile
- 7 Un intero positivo → l'input, con queste eccezioni:
 - multipli di 3 -> “Fizz”, multipli di 5 (ma non di 3) -> “Buzz”, multipli di 3 e 5 -> “FizzBuzz”
- 8 Tre interi → un array con i tre numeri in ordine (non usando un algoritmo di sorting)
- 9 Un intero → true se è maggiore di uno e primo

Loop

- 1 Due interi \rightarrow somma tutti i valori nell'intervallo chiuso, o zero. Ad es: $(1, 3) \rightarrow 6$ e $(3, 1) \rightarrow 0$
- 2 Due interi \rightarrow somma tutti i numeri pari nell'intervallo, o zero
- 3 Un intero positivo, $n \rightarrow$ un array di n stringhe contenenti solo 'x' di lunghezza crescente da 1 a n
- 4 Un numero $y \rightarrow$ la sua radice quadrata usando il metodo di Newton $x_{n+1} = (x_n + y/x_n)/2$
 - Due versioni: con epsilon predefinito (0.001) e specificato dal chiamante
- 5 Un intero \rightarrow la somma delle sue cifre
- 6 Un intero $n \rightarrow$ l'intero ottenuto concatenando il quadrato delle cifre di n . Ad es: $345 \rightarrow 91625$
- 7 Un (piccolo) intero \rightarrow il suo fattoriale
- 8 Un (piccolo) intero \rightarrow il suo Fibonacci [$\text{Fibonacci}_0 = 0, \dots, \text{Fibonacci}_6 = 8$]
 - 0, 1, 1, 2, 3, 5, 8, ...
- 9 Un (piccolo) intero $n \rightarrow$ la tavola pitagorica fino a n (array bidimensionale di interi)

Stringhe

- 1 Una stringa → la sua copia ribaltata
- 2 Una stringa → true se è un palindromo
- 3 Una stringa → la sua copia senza vocali
- 4 Una stringa s, un intero i → copia di s, senza il carattere in posizione i
- 5 Due stringhe → true se una è l'anagramma dell'altra
- 6 NIBS, numero intero binario come stringa (es: "1111" → 15)
 - a Un nibs → l'intero corrispondente
 - b Due nibs di pari lunghezza → la loro somma (bit a bit)

Array

- 1 Dato un array di interi, va ribaltato
 - Due versioni, una ritorna una copia, l'altra modifica l'input (*in place*)
- 2 Un array di interi → la media
 - ↳ La mediana – elemento centrale o media aritmetica tra i due elementi centrali
- 3 Un array di interi → il massimo
- 4 Un array di interi → l'elemento non duplicato
 - Esempio: [1,4,2,3,3,2,1] → 4
- 5 Un array di interi → i duplicati
 - Esempio: [1,4,2,3,3,2,1] → [1,2,3]
- 6 Due array di interi ordinati → un array ordinato “merged”, in $O(n)$
 - Esempio: [1,3,5,7], [2,4,6] → [1,2,3,4,5,6,7]

Altri

- `hasOnlyUnique()` // “hello” → false, “helo” → true – $O(n)$
- `singles(array)` → tutti gli elementi non duplicati
- `isArmstrong()`
 - https://it.wikipedia.org/wiki/Numero_di_Armstrong
- `isPangram()`
 - <https://it.wikipedia.org/wiki/Pangramma>
- `hammingDistance()`
 - https://it.wikipedia.org/wiki/Distanza_di_Hamming
- `acronym()` // Model View Controller → MVC
- `yahtzee(int[] dice, Category cat)`
 - <https://it.wikipedia.org/wiki/Yahtzee>

Numeri perfetti

https://it.wikipedia.org/wiki/Numero_perfetto

- `isPerfect()` // perfetto
 - La somma dei divisori propri è uguale al numero: $1 + 2 + 3 == 6$
- `isAbudant()` // abbondante
 - La somma è maggiore: $1 + 2 + 3 + 4 + 6 > 12$
- `isDeficient()` // difettivo
 - La somma è minore: $1 + 2 + 4 < 8$

Algorithms and Data Structures

- MOOC di UC San Diego
 - Massimo prodotto di una coppia di numeri
 - Input: array di interi non-negativi. Es: $\{1, 2, 3\} \rightarrow 6$
 - Ultima cifra di un numero di Fibonacci
 - $\text{Fibonacci}_{170} = 150804340016807970735635273952047185$
 - L'ultima cifra di Fibonacci_{91239} è 6
 - Calcolare il Massimo Comun Divisore (MCD)
 - $28851538, 1183019 \rightarrow 17657$
 - Calcolare il minimo comune multiplo (mcm)
 - $28851538, 1183019 \rightarrow 1933053046$