Esercitazione 8 - 16/11

Indice degli esercizi

Esercizio	Argomento
Esercizio 1	Inversione di un array
Esercizio 2	Ogni 2 è seguito da un 3
Esercizio 3	Riempire un array con numeri casuali
Esercizio 4	Riempire un array con numeri chiesti all'utente
Esercizio 5	Copiare un array: perché non basta `b = a`

Esercizio 1

Ripasso: Recap 1 · Gli array (indici, `length`, ciclo `for`) e Recap 2 · Array e funzioni (regola del nuovo array).

Dato un array `v`, scrivere un programma che costruisca un **nuovo array** `risultato` (senza modificare `v`) e lo stampi.

A. `risultato` contiene gli stessi elementi di `v` in ordine **inverso**. Ad esempio con `v = [1, 2, 3, 4]` si stampa `[4, 3, 2, 1]`.

B. In `risultato` gli elementi che si trovano in **posizione pari** (indici `0, 2, 4, ...`) sono presi in ordine inverso, mentre quelli in posizione dispari **restano dove sono**. Ad esempio:

- con `v = [1, 2, 3, 4, 5]` si stampa `[5, 2, 3, 4, 1]`
- con `v = [10, 11, 12, 13, 14, 15]` si stampa `[14, 11, 12, 13, 10, 15]`

Provate i programmi anche con un array di un solo elemento e con l'array vuoto.

Esercizio 2

Ripasso: Recap 1 · Gli array; per la variabile flag, l'esercizio di ricerca dello zero dell'esercitazione 7.

Dato un array `v`, scrivere un programma che stampi `true` se **ogni** `2` presente nell'array è **immediatamente seguito** da un `3`, `false` altrimenti. Se non ci sono `2` la risposta è `true`; un `2` in ultima posizione non è seguito da niente, quindi dà `false`.

Provate il programma con questi array:

v	cosa stampa
`[1, 2, 3, 5, 2, 3]`	`true`
`[2, 4, 3]`	`false`
`[5, 2, 3, 2]`	`false`
`[1, 5, 7]`	`true`
`[2, 2, 3]`	`false`
`[]`	`true`

Esercizio 3

Ripasso: Recap 2 · Array e funzioni; la funzione `contiene` è nell'Esercizio 11 dell'esercitazione 7.

Funzioni che **costruiscono** un array di numeri casuali rispettando una regola. Ricordate che un intero casuale tra `min` e `max` (estremi inclusi) è `Math.floor(Math.random() * (max - min + 1)) + min`.

A. Scrivere la funzione `casualiPari(n, min, max)` che restituisce un array di `n` numeri casuali **pari** compresi tra `min` e `max`.

B. Scrivere la funzione `casualiDistinti(n, min, max)` che restituisce un array di `n` numeri casuali compresi tra `min` e `max` **tutti diversi tra loro**. Suggerimento: estraete un numero con un `do ... while` finché non ne trovate uno che non è già nell'array.

Provatele, ad esempio, con `casualiPari(8, 1, 20)` e `casualiDistinti(5, 1, 10)`.

Esercizio 4

Ripasso: Recap 2 · Array e funzioni.

Funzioni che **costruiscono** un array con numeri **chiesti all'utente**. Si legge con `prompt` e si converte con `Number`, come negli esercizi precedenti.

A. Scrivere la funzione `leggiArray(n)` che chiede all'utente `n` numeri e restituisce l'array che li contiene, nell'ordine in cui sono stati inseriti.

B. Scrivere la funzione `leggiFinoAZero()` che chiede numeri **finché l'utente non inserisce `0`** e restituisce l'array dei numeri inseriti, **senza** lo zero finale. Non si sa in anticipo quanti saranno.

C. Scrivere la funzione `leggiPositivi(n)` che chiede all'utente `n` numeri **positivi**: se l'utente ne inserisce uno non positivo, il programma lo richiede finché non è valido.

Per ciascuna scrivete poi un breve programma che la usa e stampa l'array letto.

Esercizio 5

Ripasso: Recap 2 · Array e funzioni (la parte sul passaggio per riferimento).

Un array **non si copia** con l'assegnazione. Vediamo perché e come si fa.

A. Dite cosa stampa questo programma, **senza eseguirlo**. Poi spiegate con le vostre parole perché `a` è cambiato anche se abbiamo modificato solo `b`.

```
let a = [1, 2, 3];
let b = a;
b[0] = 99;
console.log(a);
console.log(b);
console.log(a == b);
```

B. Volevamo **conservare** `a` e lavorare su una copia. Scrivere la funzione `copia(v)` che restituisce un **nuovo array** con gli stessi elementi di `v`, e usarla per correggere il programma di A: dopo la modifica di `b`, `a` deve essere ancora `[1, 2, 3]`.

Ludovica Pannitto - Università di Pisa - CC BY-NC-SA 4.0