Lab 7 Black Box Testing, Unit Testing

L'obiettivo di questo laboratorio è familiarizzare con le tecniche di black box testing di piccoli moduli software. Per ciascuno dei seguenti moduli, definire i test case applicando la equivalence classes partitioning e le boundary condition.
Utilizzare la seguente struttura per documentare i test case, definendo chiaramente i criteri, le condizioni sui criteri (partizione), i test case per ciascuna partizione.
Questo laboratorio può essere svolto individualmente.

Struttura della documentazione

Criteri

ID Criterio	Descrizione
Criterio 1	C1
Criterio 2	C2
…	…

Predicati

Criterio	Predicato
Criterio 1	C1 == true
	C1 == false
Criterio 2	C2 < 0
	C2 > 0
…	…

Boundary

Criterio	Boundary
C2	C2 == 0
…	…

Classi di equivalenza e test

C1	C2	Valido / Non valido	Test case
true	< 0		T1 =
	> 0		T2 =
	0		T3 =
false	< 0		
	> 0		

Lab 7 Black Box Testing, Unit Testing

L'obiettivo di questo laboratorio è familiarizzare con le tecniche di black box testing di piccoli moduli software. Per ciascuno dei seguenti moduli, definire i test case applicando la equivalence classes partitioning e le boundary condition.
Utilizzare la seguente struttura per documentare i test case, definendo chiaramente i criteri, le condizioni sui criteri (partizione), i test case per ciascuna partizione.
Questo laboratorio può essere svolto individualmente.

Struttura della documentazione

Criteri

ID Criterio	Descrizione
Criterio 1	C1
Criterio 2	C2
…	…

Predicati

Criterio	Predicato
Criterio 1	C1 == true
	C1 == false
Criterio 2	C2 < 0
	C2 > 0
…	…

Boundary

Criterio	Boundary
C2	C2 == 0
…	…

Classi di equivalenza e test

C1	C2	Valido / Non valido	Test case
true	< 0		T1 =
	> 0		T2 =
	0		T3 =
false	< 0		
	> 0		

Esercizio 1

La funzione acceptableToEat riceve il peso in grammi di carboidrati, proteine e grassi in una porzione di cibo. Restituisce true se:

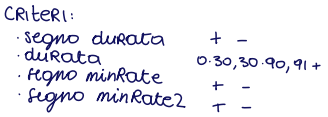
- la quantità totale di calorie è < 1000
- (carb + protein) / fat > ½

```
function acceptableToEat(carb: number, protein: number, fat: number): boolean

console.log(acceptableToEat(100, 100, 100)); // false
//(tot amount of calories = 100*4 + 100*4 + 100*9 > 1000)
console.log(acceptableToEat(1, 1, 10));     // false
console.log(acceptableToEat(1, 1, 1));      // true
```

Esercizio 2

Questa funzione calcola (in euro) il costo per il noleggio di una bicicletta, utilizzando i seguenti parametri:



- durata: minuti per cui la bicicletta è stata utilizzata
- minRate: costo al minuto, in centesimi di euro
- minRate2: costo al minuto, in centesimi di euro

Il costo è calcolato come segue: gratis per i primi 30 minuti. minRate per minuto per la prima ora eccedente i primi 30 minuti (30-90 minuti), minRate2 dopo i 90 minuti.

```
function computeFee(duration: number, minRate: number, minRate2: number): number

// Esempi:
console.log(computeFee(35, 10, 20)); // 50
console.log(computeFee(65, 10, 20)); // 350
console.log(computeFee(95, 10, 20)); // 700
```

ESERCIZIO 1:

Criteri:
- Segno calorie
- Segno proteine
- Segno grassi
- Formula 1 (c)
- Formula 2 (a)

Predicati:
- Positivo, Negativo
- Positivo, negativo
- Positivo, negativo
- Vero, Falso
- Vero, Falso

Boundaries:
- C == -1, C == 0
- C == maxInt, C == maxInt + 1
- P == -1, P == 0
- P == MI, P = MI + 1
- G == -1, G == 0
- G == 0, G == 1
- G == MI, G == MI + 1

	C	P	G	F1	F2	V	T
1	+	+	+	✓	✓	✓	T (50, 50, 20) → True
2	+	+	+	✓	X	X	T (1, 1, 9) → False
3	+	+	+	X	✓	X	T (100, 100, 100) → False
4	+	+	+	X	X	X	T (400, 100, 400) → False
5	+	+	−	NV	NV	NV	T (1, 1, -1) → ERRORE
6	...						
7	−	−	−	NV	NV	NV	T (-1, -1, -1) → ERRORE
8	+	+	+ (0)	✓	NV	NV	T (1, 1, 0) → ERRORE

Esercizio 3

Una compagnia ferroviaria offre la possibilità ai minori di 15 anni di viaggiare gratis. L'offerta è dedicata a gruppi da 2 a 5 persone che viaggiano insieme. Per essere idonei all'offerta, almeno un membro del gruppo deve avere almeno 18 anni. Se questa condizione è soddisfatta, tutti i membri sotto i 15 anni viaggiano gratis, e gli altri pagano il prezzo base.

La funzione computeFee riceve come parametri basePrice (il prezzo del biglietto), n_passengers (il numero di passeggeri del gruppo), n_over18 (il numero di passeggeri di almeno 18 anni), n_under15 (il numero di passeggeri sotto i 15 anni). Restituisce l'importo che l'intero gruppo deve spendere. Restituisce un errore se i gruppi sono composti da più di 5 persone.

```
function computeFee(basePrice: number, n_passengers: number, n_over18: number,
n_under15: number): number

// Esempi:
console.log(computeFee(20.0, 3, 0, 1)); // 60.0
console.log(computeFee(30.0, 5, 1, 2)); // 150.0
```

Esercizio 3

Una compagnia ferroviaria offre la possibilità ai minori di 15 anni di viaggiare gratis. L'offerta è dedicata a gruppi da 2 a 5 persone che viaggiano insieme. Per essere idonei all'offerta, almeno un membro del gruppo deve avere almeno 18 anni. Se questa condizione è soddisfatta, tutti i membri sotto i 15 anni viaggiano gratis, e gli altri pagano il prezzo base.

La funzione computeFee riceve come parametri basePrice (il prezzo del biglietto), n_passengers (il numero di passeggeri del gruppo), n_over18 (il numero di passeggeri di almeno 18 anni), n_under15 (il numero di passeggeri sotto i 15 anni). Restituisce l'importo che l'intero gruppo deve spendere. Restituisce un errore se i gruppi sono composti da più di 5 persone.

```
function computeFee(basePrice: number, n_passengers: number, n_over18: number,
n_under15: number): number

// Esempi:
console.log(computeFee(20.0, 3, 0, 1)); // 60.0
console.log(computeFee(30.0, 5, 1, 2)); // 150.0
```

Esercizio 3

Una compagnia ferroviaria offre la possibilità ai minori di 15 anni di viaggiare gratis. L'offerta è dedicata a gruppi da 2 a 5 persone che viaggiano insieme. Per essere idonei all'offerta, almeno un membro del gruppo deve avere almeno 18 anni. Se questa condizione è soddisfatta, tutti i membri sotto i 15 anni viaggiano gratis, e gli altri pagano il prezzo base.

La funzione computeFee riceve come parametri basePrice (il prezzo del biglietto), n_passengers (il numero di passeggeri del gruppo), n_over18 (il numero di passeggeri di almeno 18 anni), n_under15 (il numero di passeggeri sotto i 15 anni). Restituisce l'importo che l'intero gruppo deve spendere. Restituisce un errore se i gruppi sono composti da più di 5 persone.

```
function computeFee(basePrice: number, n_passengers: number, n_over18: number,
n_under15: number): number

// Esempi:
console.log(computeFee(20.0, 3, 0, 1)); // 60.0
console.log(computeFee(30.0, 5, 1, 2)); // 150.0
```