

Ex 0bb “inventory” – Test black box

A retail support system manages an inventory of items. Each item has a descriptor and the number of available items.

```
public class Item { // descriptor of items in inventory
    private String itemCode // aka bar code, unique identifier of item
    private int availability; // number of items available
    private String description; // description of item
    private String name; // name of item
}
public class Inventory{
    void addItem(Item i) throws ItemAlreadyExists // adds new descriptor
    Item searchItem (String itemCode) throws ItemNotExists; // returns item with given code

public new Item(String itemCode,int quantity); // creates a new item

    int availabilityItem (String itemCode) throws ItemNotExists; // returns availability of item
    void subtractItem (String itemCode) throws ItemNotExists, ItemNotAvailable;
        // subtracts 1 to availability
    void addQtyToItem(String itemCode, int qty_to_add);
    void subtractQtyToItem(String itemCode, int qty_to_add);

}
```

Define test cases.

Ex 1bb “Events” – Test black box

A queue of events in a simulation system receives events. Each event has a time tag. It is possible to extract events from the queue, the extraction must return the event with lower time tag.

The queue discards events with negative or null time tag.

The queue must accept at most 100.000 events.

Events with the same time tag must be merged (i.e. the second received is discarded)

```
public class EventsQueue{
    public void reset();
        // cancels all events
    public void push(int timeTag) throws InvalidTag, QueueOverflow
    public int pop() throws EmptyQueue
}
```

Define test cases

1 define equivalence classes, and related boundary condition

2 for each class, define at least one test case

Tipo timeTag (int o no)	Range di timeTag (< minit, minint a 0, 1 a maxint, oltre maxint)	Numero di eventi (0 a 100000, oltre 100000)	Time tag ripetuti	
No T(Reset() Push(“stri”); errore) T(Reset() Push(‘s’); errore)				
Si	<minint T(reset Push(minint - 10) errr)			
	Minint a zero T(reset			

	Push(-400) → InvalidTag Exception)			
	1 a maxint	0 a 100000	No T(Reset Push(100) >) T(Reset Push(100) Push(200) Pop() →100 >) T(Reset Push(0) → InvalidTag >) (boundary su time tag) T(Reset Push(1) Pop() →1 >) (boundary su time tag e su numero eventi) T(Reset Pop() → EmptyQueue >) (boundary su numero eventi) T(reset Int I =1 Repeat (100000 volte) Push(i++)	

			Pop()→ 1 Boundary su num eventi	
		Oltre 100000	T(reset Int I =1 Repeat (100001 volte) Push(i++) → queueOverflow Boundary su num eventi	
			si	

Ex 2bb “int converter” – Test black box

A function converts a sequence of chars in an integer number. The sequence can start with a ‘-’ (negative number). If the sequence is shorter than 6 chars, it is filled with blanks (to the left side).

The integer number must be in the range $\text{minint} = -32768$ to $\text{maxint} = 32767$.

The function signals an error if the sequence of chars is not allowed.

The sequence of chars must be ≤ 6 chars

- 1 define equivalence classes, and related boundary condition
- 2 for each class, define at least one test case
- 3 define class Cartesian product

int convert(string s)

“1” → “ 1” → 1

“1234” → 1234

“1234567” → error

“99999” → error

“12ff99” → error

Criterio	Classe eq	validita	Test cases
Numero di caratteri	≤ 6	Si e no	T1(“1”, 1) T3(“60000”, err)
	> 6	Non valida	T2(“1234567”, err)
La stringa rappresenta un numero ben formato	Si	Si e no	
	No	Nn valida	
range	Entro -32768 e 32767		
	Oltre -32768 e 32767	Nn valida	
segno	Negative	Un po validi un po non	
	Positive	Un po validi un po non	

Numero caratteri	String ben formnata	range	segno	Validita
<= 6	Si	Entro	Piu	S T1("1", 1) T1bis("+1", 1) T("32767", 32767) T("32768", err) T("32766", 32766)
			Meno	S T2("-1", -1) T("-32768", -32768) T("-32769", err) T("-32767", -32767)
		Oltre	Piu	N T3("60000", err)
			Meno	N T4("-60000", err)
	No	Entro	Piu	N T5("1a", err) T("", err) T(" ", err) T("12 14", err) T("- ", err)
			Meno	N T6("-1a", err)
		Oltre	Piu	N T7("",)
			Meno	N T8("-8y888", err)
> 6	Si	Entro	Piu	N
			Meno	N
		Oltre	Piu	N

			Meno	N
	No	Entro	Piu	N
			Meno	N
		Oltre	Piu	N
			Meno	N

N caratteri			
0, 1 6,7	T(“”, err) T1 T(“123456”, err) T(“000000”, 0) T(“1234567”, err)		

Ex 3bb “calendar”– Test black box

Consider a method that returns the number of days in a month, given the month and year.

```
public class MyGregorianCalendar {
    public static int getNumDaysInMonth(int month, int year){
        ...
    }
}
```

The month and the year are specified as integers. By convention, 1 represents the month of January, 2 the month of February, and so on. The range of valid inputs for the year is 0 to maxInt.

To determine leap years:

```
if year modulo 400 is 0 then leap
else if year modulo 100 is 0 then no_leap
else if year modulo 4 is 0 then leap
else no_leap
```

- 1 define equivalence classes, and related boundary condition
- 2 for each class, define at least one test case

© Bruegge – Dutoit, Object Oriented Software Engineering, 2004

Mese range	Anno range	Casi di febbraio	
Meno di 1	<0		T(-1, -1000; Err)
	0 a maxint		T(-1, 1000; Err)
	>maxint		T(-1, maxint +100; Err)
1 a 12 Senza 2	<0		T(1, -1000, err)
	0 a maxint		T(1, 1000; 31) T(4, 1000; 30)
	>maxint		
2	<0	Modulo 400	Err
		No Modulo si Modulo 100	Err
		No Modulo no Modulo 100 si Modulo 4	Err
		else	Err
	0 a maxint	Modulo 400	

		Modulo 100	
		Modulo 4	
		Else	
	>maxint	Modulo 400	
		Modulo 100	
		Modulo 4	
		Else	
Oltre 12	<0		
	0 a maxint		
	>maxint		

Ex 4bb “order integers”– Test black box

The function `int orderIntegers(int v[])` shall take as input an array of three integer numbers. The program shall output the greatest number among the elements of the array. The program shall order the elements of the array in decreasing order.

1. Identify the equivalence classes for the input
2. Identify boundary conditions
3. Write test cases (one per equivalence class)

Criteria

N elementi nel vettore (3, meno 3, piu di 3)	Tipo elemento nel vettore (intero o no)	Range di int Entro minint maxint, fuori	Segno di elementi: solo neg, solo pos, misti	Ci sono elementi uguali nel vettore
Meno di 3 T([23]; err)				
Piu di 3 T([2 3 4 5] ; err)				
Esatt 3	No interi T(['a', 'b', 'c']; err) T(['a', 2, 4]; err) T(["1", "-33", "x"]; err) T([22.0, -2.0, 33.0]; err)			
	Interi	Fuori range T([maxint +10, minint - 20, 33]; err)		
		Entro range	Solo neg	No T([-7, -66, -1]; [-1, -7, -66] ; -1)
				Si T([-66, -66, -1]; [-1, -66, -66] ; -1)
			Solo pos	No T([7, 66, 1];

				[66, 7, 1] ; 66)
				Si T([7, 66 , 7]; [66, 7, 7] ; 66)
			Misti	No T([-7, 66 , -1]; [66, -1, -7] ; 66)
				Si T([-7, 66 , -7]; [66, -7, -7] ; 66)

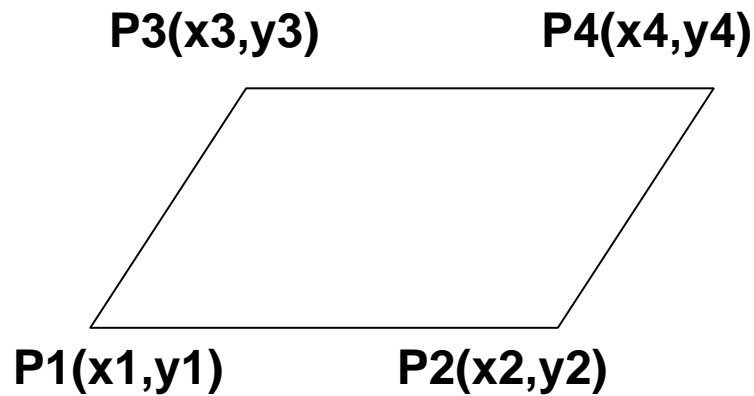
Ex 5bb “parallelogram”– Test black box

The function `parallelogram(int x1, int x2, int x3, int x4, int y1, int y2, int y3, int y4)` calculate the area of a parallelogram.

Requirements are:

- area is always strictly > 0 ;
- the parallelogram should stay in the first quadrant of the Cartesian plan;
- coordinates must have the following meaning:

In case of error or invalid input, -1 is returned.



Define equivalence classes and test cases ;

Ex 6bb “router” – Test black box

The big company “P2” has a private network in which center an internal router was placed. The router accepts packets of variable but limited length (max 1500 Bytes), and before transmitting them in the output interfaces, it performs some conformance checks:

- no more than one packet every 0.1 ms;
- destination IP must belong to the subnet 192.168.1.0;
- if source IP is 192.168.1.1, any destination is reachable;

Buffer is 1,5MB wide, packet process time is 0.2 ms .

In order to test the router, a software driver was written. The main function is:

```
int sendPacket(String srcIP, String destIp, int bytes, float time_ms);
```

The function returns the current buffer occupancy (Bytes) if packet is accepted, -1 otherwise.

Write equivalence classes and test cases.