

Esercizio 1wb – Test white box

Una coda di eventi in un sistema di simulazione riceve eventi. Ogni evento ha un tempo futuro di esecuzione (marca temporale). Le operazioni sono inserire un evento, estrarre un evento. L'estrazione deve fornire l'evento con la marca temporale piu' bassa. Eventi con tempi negativi devono essere scartati. Eventi con tempi nulli devono essere scartati.

1. Copertura NODI
2. Copertura ARCHI

Il codice e' il seguente.

```
import java.util.Iterator;
import java.util.LinkedList;

public class CodaEventi {
    private LinkedList coda;

    public CodaEventi(){
        coda = new LinkedList();
    }
    public void insert(int evento){
        int index = 0;
        int size = coda.size();
        while (index < size &&
            ((Integer)coda.get(index)).intValue() < evento){
            index++;
        }
        coda.add(index, new Integer(evento));
    }
    public int pop(){
        Object o = coda.getFirst();
        if (o != null) return ((Integer) o).intValue();
        else return -1;
    }
    public void stampa(){
        Iterator i = coda.iterator();
        int evento;
        while (i.hasNext()){
            evento = ((Integer) i.next()).intValue();
            System.out.println(evento + " ");
        }
    }
}
```

Insert

	Stato - pre	input	Stato - post	output	Nome Test
Nodi	{1}	Insert(2)	{1,2}		T1
Branch		Come sopra			T1
Condition - T T					T1
Condition - T F	{2}	Insert (1)	{1,2}		T2
Condition - F T NON POSSIBILE					
Condition - F F	{}	Insert(1)	{1}		T0
Path 0					T0
Path 1					T1
Path 2	{1,2}	Insert(3)	{1,2,3}		T4

Pop

	Stato - pre	input	Stato - post	output	Nome Test
Nodi	{}	Pop()	{}	-1	T5
Branch	{1}	Pop()	{}	1	T6

Esercizio2wb – Test white box

Una funzione converte una sequenza di caratteri in un numero intero. La sequenza puo' cominciare con un - (numero negativo). Se la sequenza e' piu' corta di 6 caratteri, viene riempita con dei blank a sinistra. L'intero rappresentato dalla sequenza deve essere compreso tra minint e maxint (minint = -32768 maxint = 32767).

La funzione deve segnalare errore se la sequenza non e' ammessa.

```
public class ConvertInt {
    public int convert(char[] str) throws Exception{
        if (str.length > 6) throw new Exception();
        int numero=0;
        int cifra; int i=0;
        if (str[0]=='-') i=1;
        for(; i<str.length; i++){
            cifra = str[i] - '\0';
            numero = numero * 10 + cifra;
        }
        if (str[0]=='-') numero = -numero;
        if (numero > 32767 || numero < -32768)
            throw new Exception();
        return numero;
    }
}
```

Coverage	Input	Output	Name
Node	"1234567"	Error	T1
	"-32769"	Error	T2
	"123"	123	T3
Branch	T1+T2+T3		
Condition	"32768"	Error	T4
Path0	""	Error	T5
	"_"	Error	T6
Path1	"1"	1	
Path2	"12"	12	
Path3	"123"	123	

Esercizio3wb – Test white box

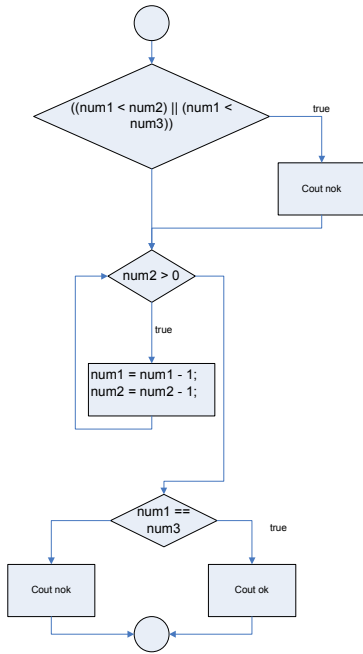
Dato la funzione C++

```
int f (int num1, int num2, int num3)
{
if ((num1 < num2) || (num1 < num3)) cout << "no ok";
while (num2 > 0) {
num1 = num1 - 1;
num2 = num2 - 1;
}
if (num1 == num3) cout << "ok";
else cout << "no ok";
}
```

Definire insiemi di test (ognuno corrispondente a un insieme di valori per le variabili in input) che coprano

- statement
- branch
- condition
- condition e branch
- path

SOLUZIONE



Critero	Input	Output	Copertura
Statement	Qualsiasi 1,2,3	Nok, Nok	100%
Branch	2,1,3	Nok, Nok	4/6 = 66%
	3,1,2	Ok	4/6 = 66%
			Combinato 100%
Decision	1,2,3 - T T nodo1	Nok, Nok	
	3,2,1 - F F nodo1	ok	
	1,2,0 - T F	Nok, nok	
	1,0,2 - F T	Nok, nok	Combinato 100% branch 100% nodi 100% decisioni
Path (2-coverage)	Zero cicli in while, 4 path -1,0,3 dx, sx -1,0, -1 dx, dx -1, -2, -3 sx sx -1, -2, -1 sx dx 1 ciclo in while, 4 path sommare 1 a tutti 2 cicli in while sommare 2 a tutti		

Esercizio4wb – white box

```
1 float homeworkAverage(float[] scores) {
2   float min = 99999;
3   float total = 0;
4   for (int i = 0; i < scores.length; i++){
5     if (scores[i] < min)
6       min = scores[i];
7     total += scores[i];
8   }
9   total = total - min;
10  return total / (scores.length - 1);
11 }
```

La funzione calcola la media di un array.

Definire test case per copertura di

- statement
- branch
- condition
- condition e branch
- path

Esercizio5wb – Test white box

```
1 int order(int v[3]){
2     int tmp;
3     if(v[0]<v[1]){
4         tmp = v[0];
5         v[1] = v[1];
6         v[1] = tmp;
7     }
8     if(v[1]<v[2]){
9         tmp = v[0];
10        v[1] = v[2];
11        v[2] = tmp;
12    }
13    return v[0];
14 }
```

La funzione ordina tre interi, ricevuti nell'array, e rende il maggiore

Definire test case per copertura di

- statement
- branch
- condition
- condition e branch
- path