

SCHEMA ESAME - ESERCIZIO 1

```
public class Pioggia {  
    int[] pioggia = { 0, 0, 0, 0 };  
    boolean tropicale() {  
        1 int somma = 0;  
        2 for (int i : pioggia) {  
            3 if (i < 10)  
                4 return false;  
            5 somma += i;  
            6 if (somma > 30 && i >= 15)  
                7 return true;  
        }  
        8 return false;  
    }  
}
```

1.1 : Istruzioni

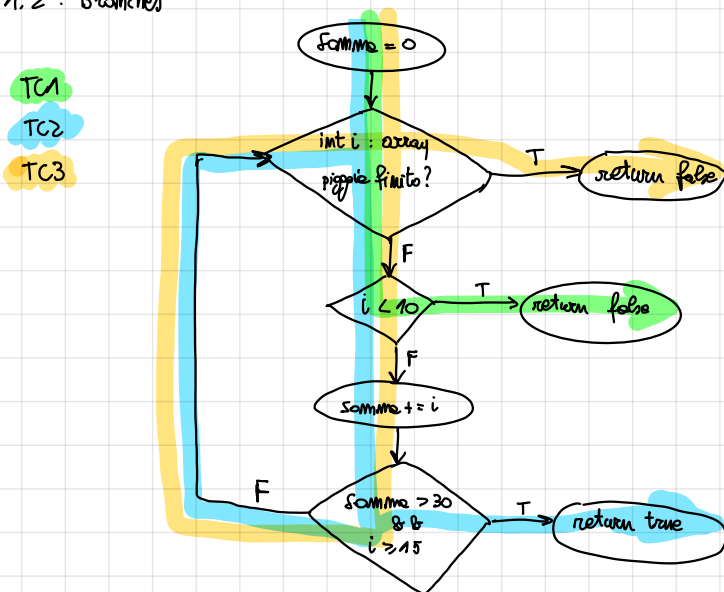
TC1 : pioggia = { 9, 0, 0, 0 } Copre 1, 2, 3, 4
TC2 : pioggia = { 12, 12, 12, 16 } Copre 1, 2, 3, 5, 6, 7
TC3 : pioggia = { 10, 10, 10, 10 } Copre 1, 2, 3, 5, 6, 8

Converto in simile JUnit

@Test

```
public void TestIstruzioni() {  
    Pioggia p = new Pioggia();  
    p.pioggia = new int[] { 9, 0, 0, 0 };    (TC1)  
    assertEquals(false, p.tropicale());  
    p.pioggia = new int[] { 12, 12, 12, 16 };    (TC2)  
    assertEquals(true, p.tropicale());  
    p.pioggia = new int[] { 10, 10, 10, 10 };    (TC3)  
    assertEquals(false, p.tropicale());  
}
```

1.2 : Branches



I casi di test TC1, TC2, TC3 coprono tutti i branches. Per JUnit basta il metodo TestIstruzioni().

1.3 : MCDC

$i < 10$	
T	TC1
F	TC2

$Summa > 30 \ \&\& \ i \geq 15$	
T	T (TC2: $Summa = 52, i = 16$)
F	(T) (TC4)
T	T
T	F (TC2: $Summa = 36, i = 12$)

Anche se c'è Easy eval. e il true non viene valutato, aggiungiamo per completezza il caso di test TC4.

TC4: piggie: {16, 0, 0, 0} ($Summa = 16, i = 16$)

Per coprire l' MCDC, in aggiunta a TestStrutture(), aggiungiamo anche il metodo:

@Test

```
public void TestMcdc() {
    Piggia p = new Piggia();
    p.piggia = new int[] {16, 0, 0, 0};
    assertEquals(false, p.tropicale());
}
```