

```
1 boolean racer_disqualified(int times[3], int winner_times[3], int n_penalties, int* penalties) {
2
3     bool disqualified = false;
4     int i;
5     int max_time;
6     int tot_penalties = 0;
7
8     for (i = 0; i < n_penalties; i++) {
9         tot_penalties += penalties[i];
10        if (penalties[i] > 100)
11            disqualified = true;
12    }
13
14    if (tot_penalties > 100 || n_penalties > 5)
15        disqualified = true;
16
17    for (i=0; i<3; i++) {
18
19        max_time = winner_times[i] * 1.5;
20
21        if (times[i] > max_time)
22            disqualified = true;
23    }
24    return disqualified;
25 }
```

For node coverage

$N_{\text{penalties}} \geq 1$

one penalty > 100

One time higher than the max time

TC1

Times = {20,20,10}

Winner_times = {10,10,10}

$N_{\text{penalties}}=2$

Penalties={150, 50}

TC2

Times={20,20,10}

Winner_times={10,10,10}

$N_{\text{penalties}}=0$

Penalties={}

For edge coverage

TC2

Times={20,20,10}

Winner_times={10,10,10}

N_penalties=0

Penalties={}

For MC

1 TC with tot_penalties > 100 and n_penalties > 5 [TT] -> TC3 (below)

1 TC with tot_penalties > 100 and n_penalties <= 5 [TF] -> TC1

1 TC with tot_penalties <= 100 and n_penalties > 5 [FT] -> TC4 (below)

1 TC with tot_penalties <= 100 and n_penalties <= 5 [FF] -> TC2

TC3

Times={20,20,10}

Winner_times={10,10,10}

N_penalties=6

Penalties= {50,50,50,50,50,50}

TC4

Times={20,20,10}

Winner_times={10,10,10}

N_penalties=6

Penalties= {5,5,5,5,5,5}

For loop

1 TC with 0 iterations -> TC2

1 TC with 1 iteration -> TC5 (below)

1 TC with 2+ iterations -> any of TC1, TC3, TC4

TC5

Times={20,20,10}

Winner_times={10,10,10}

N_penalties=1

Penalties= {50,5}

Coverage type	Number of test cases needed to obtain 100% coverage	Coverage obtained with test cases defined (%)	Test cases defined
Node	1	100%	TC1
Edge	2	100%	TC1, TC2
Multiple condition line 14	4	100%	TC1, TC2, TC3, TC4
Loop line 8	3	100%	TC1, TC2, TC5
Path	$2^{(4+np)}$	Coverage can be obtained with automated test cases generation	-



