

```
1 #lang racket
3
4 ; Task 1
5 ; comment
6 (displayln "Hello, world!")
7
8
9 ; Task 2
10 (displayln "Winter is coming")
11
12
13 ; Task 3
14 (- 128 37)
15
16
17 ; Task 4
18 (- 10 100 12 18)
19
20
21 ; Task 5
22 (- 100 34 22 (- (+ 5 3) 10))
23
24 ; Task 6
25 (+ 4 (- 2 (+ (* 3 5) (/ 8 7))))
26
27 ; Task 7
28 (define members-number 100)
29
30 (displayln members-number)
31
32 ; Task 8
33 (define (cube n) (* n (* n n)))
34 (cube 3)
35
```

```
36 ; Task 9
37 (define number
38   ((lambda (a b) (/ (+ a b) 2))
39    2 4))
40
41 (displayln number)
42
43 ; Task 10
44 (define (sum-of-squares x y) (+ (* x x) (* y y)))
45 (sum-of-squares 2 3)
46
47 ; Task 11
48 (provide telephone)
49
50 (define telephone "samsung")
51
52 ; Task 12
53 (define (square-of-sum x y)
54   (let ([sum (+ x y)])
55     (* sum sum)))
56
57 (square-of-sum 2 3)
58
59 ; Task 13
60 (define (same-parity? x y) (equal? (even? x) (even? y)))
61
62 (same-parity? 3 7) ; #t
63 (same-parity? 4 8) ; #t
64 (same-parity? 4 7) ; #f
65 (same-parity? 3 10) ; #f
66
```

```

6 ; Task 14
7 (define (sentence-type text) (if (equal? (string-upcase text) text)
8     "cry"
9     "common"))
10
11 (sentence-type "HOW ARE YOU?") ; "cry"
12 (sentence-type "Hello, world!") ; "common"
13
14 ; Task 15
15 (define (say-boom v) (when (equal? v "go") "Boom!"))
16
17 (say-boom "hey")
18 (say-boom "go") ; "Boom!"
19
20 ; Task 16
21 (define (humanize-permission v)
22     (case v
23         [( "x" ) "execute"]
24         [( "w" ) "write"]
25         [( "r" ) "read"]))
26
27 (humanize-permission "x") ; execute
28 (humanize-permission "r") ; read
29 (humanize-permission "w") ; write
30
31 ; Task 17
32 (define (programmer-level points-count)
33     (cond
34         [( < points-count 10 ) "junior"]
35         [(and ( >= points-count 10 ) (< points-count 20)) "middle"]
36         [else "senior"]))
37
38 (programmer-level 10) ; middle
39 (programmer-level 0) ; junior
40 (programmer-level 40) ; senior
41
42

```

```

Language: Haskell, Main: REPL
Hello, world!
Winter is coming
91
-120
46
-101/7
100
27
3
13
25
#t
#t
#f
#f
"cry"
"common"
"Boom!"
"execute"
"read"
"write"
"middle"
"junior"
"senior"
>

```