Metody Programowania lista 1

Szymon Kopa

20 lutego 2018

1 Ćwiczenie 1

1.1

```
>10
10
```

1.2

```
>(+ 5 3 4)
12
```

1.3

```
>(- 9 1)
8
```

1.4

```
>(/ 6 2)
3
```

1.5

```
>(+ (* 2 4) (- 4 6) )

6
```

1.6

```
>(define a 3)
>( define b (+ a 1) )
>(+ a b (* a b ) )

19
```

1.7

```
>(= a b )

#f
```

1.8

```
>( if ( and (> b a ) (< b (* a b ) ) ) b a)

4
```

1.9

```
>( cond [(= a 4) 6]
[(= b 4) (+ 6 7 a ) ]
[ else 25])

16
```

1.10

```
>(+ 2 ( if (> b a ) b a ) )

6
```

1.11

```
>(* ( cond [( > a b ) a ]
[(< a b ) b ]
[ else
-1])
(+ a 1) )

16
```

2 Ćwiczenie 2

$$\frac{5+4+(2-(3-(6+\frac{4}{5})))}{3(6-2)(2-7)}$$

```
    (/ ({+ 5 4 (- 2 {- 3 (+ 6 {/ 4 5})})}) {* 3 (- 6 2 ) (- 2 7 )})
```

3 Ćwiczenie 3

4 Ćwiczenie 4

```
#lang racket
(define (sum-of-squares a b)
  (+ (* a a) (* b b)))
(define (max a b)
  (cond [(> a b) a]
        [else b]))
(define (sum-of-squares-of-2-largest a b c)
  (cond [(> a b) (sum-of-squares a (max b c))]
        [else (sum-of-squares b (max a c))]))
```