

1. Funcția `id`:

`id :: a -> a`

2. Funcția `const`:

`const :: a -> b -> a`

3. Funcția `flip`:

`flip :: (a -> b -> c) -> b -> a -> c`

4. Funcția `(.)` (compoziție de funcții):

`(.) :: (b -> c) -> (a -> b) -> a -> c`

5. Funcția `(\$)` (aplicare de funcții)

`($) :: (a -> b) -> a -> b`

6. Funcția `map`:

`map :: (a -> b) -> [a] -> [b]`

7. Funcția `filter`:

`filter :: (a -> Bool) -> [a] -> [a]`

8. Funcția `foldr`:

`foldr :: (a -> b -> b) -> b -> [a] -> b`

9. Funcția `foldl`:

`foldl :: (b -> a -> b) -> b -> [a] -> b`

10. Funcția `zip`:

`zip :: [a] -> [b] -> [(a, b)]`

11. Funcția `zipWith`:

`zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]`

12. Funcția `concat`:

`concat :: [[a]] -> [a]`

13. Funcția `head`:

`head :: [a] -> a`

14. Funcția `tail`:

`tail :: [a] -> [a]`

15. Funcția `init`:

`init :: [a] -> [a]`

16. Funcția `last`:

`last :: [a] -> a`

17. Funcția `reverse`:

`reverse :: [a] -> [a]`

18. Funcția `length`:

`length :: [a] -> Int`

19. Funcția `take`:

`take :: Int -> [a] -> [a]`

20. Funcția `drop`:

`drop :: Int -> [a] -> [a]`

21. Funcția `elem`:

`elem :: Eq a => a -> [a] -> Bool`

22. Funcția `notElem`:

`notElem :: Eq a => a -> [a] -> Bool`

23. Funcția `maximum`:

`maximum :: Ord a => [a] -> a`

24. Funcția `minimum`:

`minimum :: Ord a => [a] -> a`

25. Funcția `sum`:

`sum :: Num a => [a] -> a`

26. Funcția `product`:

`product :: Num a => [a] -> a`

27. Funcția `elemIndex`:

`elemIndex :: Eq a => a -> [a] -> Maybe Int`

28. Funcția `elemIndices`:

`elemIndices :: Eq a => a -> [a] -> [Int]`

29. Funcția `find`:

`find :: (a -> Bool) -> [a] -> Maybe a`

30. Funcția `partition`:

`partition :: (a -> Bool) -> [a] -> ([a], [a])`

EXERCITII

Exemplu 1:

-- Exemplu 1

`f1 x y = (x > y, x + y)`

Sintetizare:

`f1 :: Ord a => a -> a -> (Bool, a)`

`x :: Ord a => a`

`y :: Ord a => a`

`(>) :: Ord a => a -> a -> Bool`

`(+) :: Num a => a -> a -> a`

Exemplu 2:

-- Exemplu 2

`f2 x y = x + (head y)`

Sintetizare:

`f2 :: Num a => a -> [a] -> a`

`x :: Num a => a`

y :: [a]

head :: [a] -> a

(+) :: Num a => a -> a -> a

Exemplu 3:

-- Exemplu 3

f3 f g x = f (g x) (g (x + 1))

Sintetizare:

f3 :: (a -> b -> c) -> (d -> a) -> d -> c

f :: a -> b -> c

g :: d -> a

x :: d

(+) :: Num d => d -> d -> d

Exemplu 4:

-- Exemplu 4

f4 xs = length xs > 0

Sintetizare:

f4 :: [a] -> Bool

xs :: [a]

length :: [a] -> Int

(>) :: Ord a => Int -> Int -> Bool

Exemplu 5:

f5 x = [x, x + 1, x + 2]

Sintetizare:

f5 :: Num a => a -> [a]

x :: Num a => a

(+) :: Num a => a -> a -> a

Exemplu 6:

f6 x y = x ++ y

Sintetizare:

f6 :: [a] -> [a] -> [a]

x :: [a]

y :: [a]

(++) :: [a] -> [a] -> [a]

Exemplu 7:

f7 xs = filter (\x -> x > 0) xs

Sintetizare:

f7 :: (Num a, Ord a) => [a] -> [a]

xs :: (Num a, Ord a) => [a]

filter :: (a -> Bool) -> [a] -> [a]

(>) :: Ord a => a -> a -> Bool

Exemplu 8:

f8 xs = map (\x -> x * x) xs

Sintetizare:

f8 :: Num b => [b] -> [b]

xs :: Num b => [b]

map :: (a -> b) -> [a] -> [b]

(*) :: Num a => a -> a -> a

Exemplu 9:

f9 f g xs = foldr (\x acc -> f (g x) acc) [] xs

Sintetizare:

f9 :: (b -> a -> b) -> (c -> a) -> [c] -> b

f :: b -> a -> b

g :: c -> a

xs :: [c]

foldr :: (a -> b -> b) -> b -> [a] -> b

Exemplu 10:

f10 f g x = f (g x)

Sintetizare:

f10 :: (a -> b) -> (c -> a) -> c -> b

f :: a -> b

g :: c -> a

x :: c

Exemplu 11:

f11 xs = reverse (tail xs)

Sintetizare:

f11 :: [a] -> [a]

xs :: [a]

reverse :: [a] -> [a]

tail :: [a] -> [a]

Exemplu 12:

f12 f g x y = f (g (x, y))

Sintetizare:

f12 :: ((a, b) -> c) -> (d -> (a, b)) -> d -> c

f :: (a, b) -> c

g :: d -> (a, b)

x :: d

y :: d

Exemplu 13:

f13 x = x == reverse x

Sintetizare:

f13 :: Eq a => [a] -> Bool

x :: Eq a => [a]

(==) :: Eq a => a -> a -> Bool

reverse :: [a] -> [a]

Exemplu 14:

f14 f g x = f (g x) x

Sintetizare:

f14 :: (a -> b -> c) -> (d -> a) -> d -> c

f :: a -> b -> c

g :: d -> a

x :: d

Exemplu 15:

f15 x y = zipWith (\a b -> (a, b)) x y

Sintetizare:

f15 :: [a] -> [b] -> [(a, b)]

x :: [a]

y :: [b]

zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]

Exemplu 16:

f16 xs = sum [1 | _ <- xs]

Sintetizare:

f16 :: Num a => [b] -> a

xs :: [b]

sum :: Num a => [a] -> a

Exemplu 17:

f17 xs = foldl (\acc x -> if even x then x:acc else acc) [] xs

Sintetizare:

f17 :: Integral a => [a] -> [a]

xs :: Integral a => [a]

foldl :: (b -> a ->

b) -> b -> [a] -> b

even :: Integral a => a -> Bool

Exemplu 18:

f18 x y = zip (x ++ [y]) [y]

Sintetizare:

f18 :: [a] -> a -> [(a, a)]

x :: [a]

y :: a

zip :: [a] -> [b] -> [(a, b)]

(++) :: [a] -> [a] -> [a]

Exemplu 19:

f19 x y = replicate x y

Sintetizare:

f19 :: Int -> a -> [a]

x :: Int

y :: a

replicate :: Int -> a -> [a]

Exemplu 20:

f20 x = map (\y -> y * y) x

Sintetizare:

f20 :: Num b => [b] -> [b]

x :: Num b => [b]

map :: (a -> b) -> [a] -> [b]

(*) :: Num a => a -> a -> a

Exemplu 21:

f21 x = map (\y -> (y, y * y)) x

Sintetizare:

f21 :: Num b => [a] -> [(a, b)]

x :: [a]

map :: (a -> b) -> [a] -> [b]

(*) :: Num a => a -> a -> a

Exemplu 22:

f22 f x y = f (x + y)

Sintetizare:

f22 :: (Num a, Num b) => (a -> b) -> a -> a -> b

f :: Num a => a -> b

x :: Num a => a

y :: Num a => a

(+) :: Num a => a -> a -> a

Exemplu 23:

f23 f x y = f (f x) (f y)

Sintetizare:

f23 :: (a -> b) -> a -> a -> b

f :: a -> b

x :: a

y :: a

Exemplu 24:

f24 x = zip x (tail x)

Sintetizare:

f24 :: [a] -> [(a, a)]

x :: [a]

zip :: [a] -> [b] -> [(a, b)]

tail :: [a] -> [a]

Exemplu 25:

f25 f g x = (f x, g x)

Sintetizare:

f25 :: (a -> b) -> (a -> c) -> a -> (b, c)

f :: a -> b

g :: a -> c

x :: a

Exemplu 1:

f1 x = filter (\y -> y > 0) x

Sintetizare:

f1 :: (Num a, Ord a) => [a] -> [a]

x :: (Num a, Ord a) => [a]

filter :: (a -> Bool) -> [a] -> [a]

(>) :: Ord a => a -> a -> Bool

Exemplu 2:

f2 f x y = (f x, f y)

Sintetizare:

f2 :: (a -> b) -> a -> a -> (b, b)

f :: a -> b

x :: a

y :: a

Exemplu 3:

f3 x y = x : y : []

Sintetizare:

f3 :: a -> a -> [a]

x :: a

y :: a

Exemplu 4:

f4 f g x = f (g (g x))

Sintetizare:

f4 :: (a -> b) -> (c -> a) -> c -> b

f :: a -> b

g :: c -> a

x :: c

Exemplu 5:

f5 x = x ++ reverse x

Sintetizare:

f5 :: [a] -> [a]

x :: [a]

(++) :: [a] -> [a] -> [a]

reverse :: [a] -> [a]

Exemplu 6:

f6 f x y = f (x + y) (x - y)

Sintetizare:

f6 :: (Num a, Num b) => (a -> a -> c) -> a -> b -> c

f :: (Num a, Num b) => a -> a -> c

x :: (Num a, Num b) => a

y :: (Num a, Num b) => b

(+) :: Num a => a -> a -> a

(-) :: Num a => a -> a -> a

Exemplu 7:

f7 x y = x : y : []

Sintetizare:

f7 :: a -> a -> [a]

x :: a

y :: a

Exemplu 8:

f8 f g x y = f (g (x, y))

Sintetizare:

f8 :: ((a, b) -> c) -> (d -> (a, b)) -> d -> c

f :: (a, b) -> c

g :: d -> (a, b)

x :: d

y :: d

Exemplu 9:

f9 x y = replicate x y

Sintetizare

f9 :: Int -> a -> [a]

x :: Int

y :: a

replicate :: Int -> a -> [a]

Exemplu 10:

f10 f x y = (f x y, f y x)

Sintetizare:

f10 :: (a -> b -> c) -> a -> b -> (c, c)

f :: a -> b -> c

x :: a

y :: b

Exemplu 11:

f11 f g x y = f (g x) (g y)

Sintetizare:

f11 :: (a -> b -> c) -> (d -> a) -> d -> d -> c

f :: a -> b -> c

g :: d -> a

x :: d

y :: d

Exemplu 12:

f12 f x y = f (f x) (f y)

Sintetizare:

f12 :: (a -> a -> b) -> a -> a -> b

f :: a -> a -> b

x :: a

y :: a

Exemplu 13:

f13 f x y = f (f x y) y

Sintetizare:

f13 :: (a -> b -> c) -> a -> b -> c

f :: a -> b -> c

x :: a

y :: b

Exemplu 14:

f14 f x = f (f x)

Sintetizare:

f14 :: (a -> a -> b) -> a -> b

f :: a -> a -> b

x :: a

Exemplu 15:

f15 f x = f (f (f x))

Sintetizare:

f15 :: (a -> a) -> a -> a

f :: a -> a

x :: a

Exemplu 16:

f16 f g x y = f (g x y)

Sintetizare:

f16 :: (a -> b -> c) -> (d -> e -> a) -> d -> e -> c

f :: a -> b -> c

g :: d -> e -> a

x :: d

y :: e

Exemplu 17:

f17 f g x y = f (g x) (g y)

Sintetizare:

f17 :: (a -> b -> c) -> (d -> a) -> d -> d -> c

f :: a -> b -> c

g :: d -> a

x :: d

y :: d

Exemplu 18:

f18 f x y = f (f x) (f y)

Sintetizare:

f18 :: (a -> b -> c) -> (a -> a) -> a -> a -> c

f :: a -> b -> c

x :: a -> a

y :: a

Exemplu 19:

f19 f x y = f (f (x, y))

Sintetizare:

f19 :: ((a, b) -> c) -> a -> b -> c

f :: (a, b) -> c

x :: a

y :: b

Exemplu 20:

f20 f g x y = f (g x) (g y)

Sintetizare:

f20 :: (a -> b -> c) -> (d -> a) -> (e -> a) -> d -> e -> c

f :: a -> b -> c

g :: d -> a

x :: d

y :: e

Exemplu 21:

f21 f g x y = f (g x y) (g y x)

Sintetizare:

$f_{21} :: (a \rightarrow a \rightarrow b) \rightarrow (c \rightarrow c \rightarrow a) \rightarrow c \rightarrow c \rightarrow b$

$f :: a \rightarrow a \rightarrow b$

$g :: c \rightarrow c \rightarrow a$

$x :: c$

$y :: c$

Exemplu 22:

$f_{22} f g x y = f (g x y)$

Sintetizare:

$f_{22} :: (a \rightarrow b \rightarrow c) \rightarrow (d \rightarrow e \rightarrow a) \rightarrow d \rightarrow e \rightarrow c$

$f :: a \rightarrow b \rightarrow c$

$g :: d \rightarrow e \rightarrow a$

$x :: d$

$y :: e$

Exemplu 23:

$f_{23} f x y = f (x, y)$

Sintetizare:

$f_{23} :: ((a, b) \rightarrow c) \rightarrow a \rightarrow b \rightarrow c$

$f :: (a, b) \rightarrow c$

$x :: a$

$y :: b$

Exemplu 24:

$f_{24} f x y = f (f x y) (f y x)$

Sintetizare:

$f_{24} :: (a \rightarrow a \rightarrow b) \rightarrow a \rightarrow a \rightarrow b$

$f :: a \rightarrow a \rightarrow b$

$x :: a$

$y :: a$

Exemplu 25:

$f_{25} f g x y = f (g x) (g y)$

Sintetizare:

$f_{25} :: (a \rightarrow b \rightarrow c) \rightarrow (d \rightarrow a) \rightarrow (e \rightarrow b) \rightarrow d \rightarrow e \rightarrow c$

$f :: a \rightarrow b \rightarrow c$

$g :: d \rightarrow a$

$x :: d$

$y :: e$