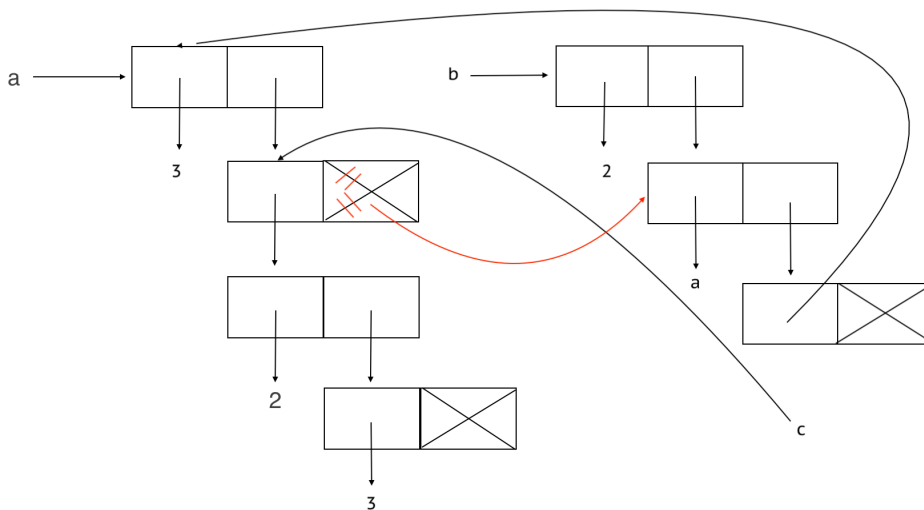


Демовариант контрольной работы по фп

2.I

```
(define-syntax when
  (syntax-rules ()
    ((when test)
      (let ((t test))
        (if t t
            #f)
      )
    )
    ((when test expr1)
      (if test expr1
          #f)
      )
    )
    ((when test expr1 expr2 ...)
      (if test (begin expr1 (when #t expr2 ...))
          #f)
      )
    )
  )
)
```

2.V.



2.II

```
(define (reverse1 lst)
  (foldl (lambda (x y) (cons x y))
    null
    lst)
)
(define (reverse2 lst)
  (foldr (lambda (x y) (append y (list x)))
    null
    lst)
)
```

)
)

2.III.

$(\lambda a. (\lambda c. ((\lambda b. (c\ a)) ((\lambda b. b\ b) (\lambda b. b\ b\ a)))) y\ z \rightarrow \alpha$ // $b \rightarrow m$

$(\lambda a. (\lambda c. ((\lambda b. (c\ a)) ((\lambda m. m\ m) (\lambda b. b\ b\ a)))) y\ z \rightarrow \alpha$ // $b \rightarrow k$

$(\lambda a. (\lambda c. ((\lambda b. (c\ a)) ((\lambda m. m\ m) (\lambda k. k\ k\ a)))) y\ z \rightarrow \beta$

$(\lambda c. ((\lambda b. (c\ y)) ((\lambda m. m\ m) (\lambda k. k\ k\ y)))) z \rightarrow \beta$

$((\lambda b. (z\ y)) ((\lambda m. m\ m) (\lambda k. k\ k\ y))) \rightarrow \beta$

ответ: нормальная форма есть: $(z\ y)$

2.IV.

```
#lang scheme/base
```

```
(define (nthbit n)
  (let loop ((sum 0) (adding 2))
    (cond ((= sum n) 1)
          ((< sum n) (loop (+ sum adding) (+ adding 1)))
          (else 0))
  )
)
```