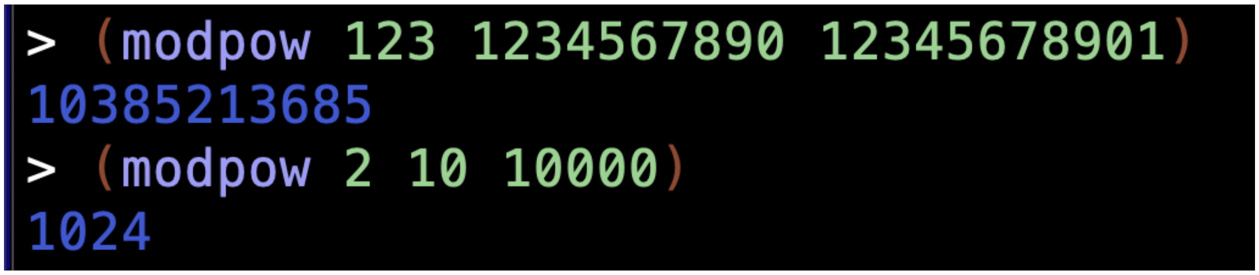
Heimadæmi 5 - Forritunarmál (Hópverkefni)

A team

September 2023

1.

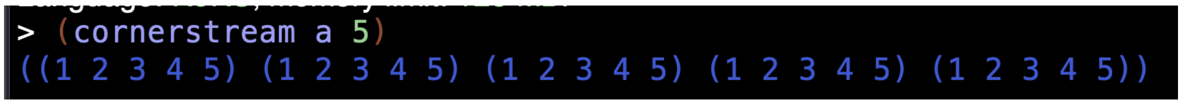
```scheme
;; Usage: (modpow p q r)
;; Pre: p, q and r are integers,
;; q > 0, 0 <= p < r.
;; Value: (p^q)%r
(define (modpow p q r)
  ;; Usage: (helper p q s)
  ;; Pre: p, q and s are integers,
  ;; 0 <= p < r , q >= 0.
  ;; Value: The integer (s*(p^q))%r.
  (define (helper p q s)
    (if (= q 0)
        s
        (if (= (remainder q 2 ) 0 )
            (helper (remainder (* p p)r)
                    (quotient q 2) s
            )
            (helper (remainder (* p p)r)
                    (quotient q 2)
                    (remainder (* p s)r)
            )
        )
    )
  )
  (helper p q 1)
)
```



```
> (modpow 123 1234567890 12345678901)
10385213685
> (modpow 2 10 10000)
1024
```

2.

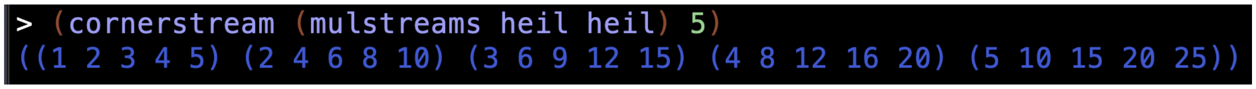
```scheme
;;Usage: (cornerstream s n)
;;Pre: s is an infinite stream of
;;infinite streams,
;;s=[[x11 x12 ...],[x21 x22 ...] ...].
;;n isan integer, n>=0.
;;Value: The list
;;((x11 x12 ... x1n)
;;(x21 x22 ... x2n)
;;...
;;(xn1 xn2 ... xnn)
;;)
(define (cornerstream s n)
  (stream-list (stream-map (lambda (x) (stream-list x n)) s)
  n)
)
```



```
> (cornerstream a 5)
((1 2 3 4 5) (1 2 3 4 5) (1 2 3 4 5) (1 2 3 4 5) (1 2 3 4 5))
```

3.

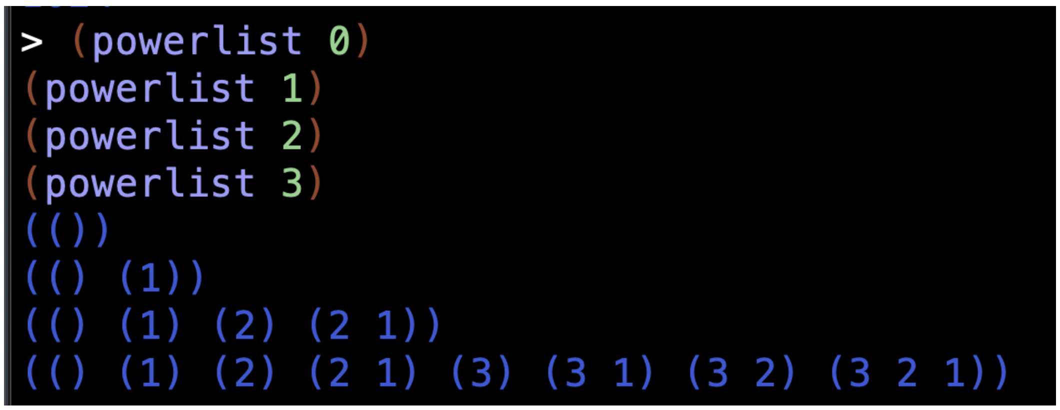
```scheme
;; Notkun: (mulstreams x y)
;; Fyrir: x and y are infinite streams of numbers,
;; x=[x1 x2 x3 ...].
;; y=[y1 y2 y3 ...].
;; Gildi: An infinite stream of infinite streams
;; of numbers, namely
;; [[x1*y1 x2*y1 x3*y1 ...]
;; [x1*y2 x2*y2 x3*y2 ...]
;; [x1*y3 x2*y3 x3*y3 ...]
;; .
;; .
;; .
;; ]
(define (mulstreams x y)
  (stream-map (lambda (a)
                (stream-map (lambda (b) (* b a)) x)
              )
  y)
)
```



```
> (cornerstream (mulstreams heil heil) 5)
((1 2 3 4 5) (2 4 6 8 10) (3 6 9 12 15) (4 8 12 16 20) (5 10 15 20 25))
```

4.

```
;;Use: (powerlist n)
;;Pre: n is an integer, n >=0.
;;Value: The list (y1 y2 y3 ...)
;;that contains all lists that can be
;;have zero or more
;;values from {1, ..., n}
;;and concatenating them in a list in
;;descending order.
(define (powerlist n)
  (if (= n 0)
      '(( ))
      (let ((pl (powerlist (- n 1 ))))
        (append pl
                (map (lambda (x) (cons n x))
                      pl
                )
        )
      )
  )
)
```



```
> (powerlist 0)
(powerlist 1)
(powerlist 2)
(powerlist 3)
(())
(() (1))
(() (1) (2) (2 1))
(() (1) (2) (2 1) (3) (3 1) (3 2) (3 2 1))
```