```
/*
Temperature flow

Generates a list of temperatures, 0 <= T <= 100, and iterates heat flow through
the list.  The left bound of the list is 40, right bound is 20.  During 50
steps, the heat distributes evenly throughout the list from 40 downto 20.

Usage: temp <seed for random>
*/
#include <lambda.h>
using namespace lambda;

FUN(mapenv,T f,T leftbound,T rightbound,T list){
    let left=leftbound;
    let current=head (list);
    let rest=drop (1) (list);
    let right=choose (rightbound) (head (rest)) (isempty (rest));

    let mapfirst=(f (left) (current) (right));
    let maprest=(mapenv (f) (current) (rightbound) (rest));
    return choose
        (end)
        (front (mapfirst) (maprest))
        (isempty (list));
}

FUN(temp_loc,T left,T cur,T right){
    return divide (add (add (left) (mult (cur) (4))) (right)) (6);
}

FUN(temp_step,T leftbound,T rightbound,T curtemp){
    return mapenv (temp_loc) (leftbound) (rightbound) (curtemp);
}

MAIN(T args){
    let list=take (20) (randomRs (0) (100) (mkStdGen (head (args))));
    return
        printmatrix (take (50) (iterate (temp_step (40) (20)) (list)));
}
```