1. You are to make a procedure, (map-reverse-order proc lst) which takes a list and a procedure and produces a map in which the procedure is applied to the list in REVERSE ORDER.
   1. Examples:
      1. (map-reverse-order (lambda (a) +1) ‘(1 2 3 4 5 6 7 8 9 )) -> (10 9 8 7 6 5 4 3 2)
2. You are to make a procedure, (special lst) which takes a list of both numbers and symbols and returns the sum of: if the number is even or zero, add it to (special rest of lst), if odd, multiply, USE CALL/CC
   1. Examples:
      1. (special '(4 4 6 1 )) -> 4+(4+(6+(1\* (0))) = 14
      2. (special '(4 4 6 3 2)) -> 4+(4+(6+(3\*(2+(0))))) =20
      3. (special ‘(3 3 2 3 6 5 4 3 8 9 0 )) -> 3\*(3\*(2+(3\*(6+5\*(4+3\*(8+(9\*(0+(0))))))))) = 3960
      4. (special ‘(3 a b c d e f 3 2 7 8 a h c s o 0 3 4 8 6 5 6 0 3 p l c) -> 3\*(3\*(2+(7\*(8+(0+(3\*(4+(8+(6+(5\*(6+(0+(3\*(0)))))))))))))) = 9594
3. Translate this code to cps with no call/cc.   
   