Task 1.1

Let's split it up

Starting with ('a → 'b → 'b) it is easy to see what they corresponds to:

x :: xs → f x (foldBack f xs e)

x ↑ 'a

f xs e → 'b

f x (foldBack f xs e) → 'b

Next is xlst which is 'a list.

Then we return e if xlst = [], e is 'b

Lastly the output of the method is 'b

fold (fun a x → x − 2 · a) 0 [1; 2; 3] = 3

x = 1
xs = [2; 3]
f 0 1 = 1 − 2 · 0 = 1
fold (fun a x → 1 − 2 · 0) 1 [2; 3]

x = 2
xs = [3]
f 1 2 = 2 − 2 · 1 = 0
fold (fun a x → 2 − 2 · 1) 0 [3]

x = 3
xs = []
f 0 3 = 3 − 2 · 0 = 3
fold (fun a x → 3 − 2 · 0) 3 []

[] = 3

foldBack (fun x a → x - 2a) [1;2;3] 0 ⇝ 9

f = fun x a → x - 2a
e = 0
x = 1
xs = [2;3]
f 1 (foldBack f [2;3] 0) = f 1 -4 = 1 - 2·(-4) = 9

e = 0
x = 2
xs = [3]
f 2 (foldBack f [3] 0) = f 2 3 = 2 - 2·3 = -4

e = 0
x = 3
xs = []
f 3 (foldBack f [] 0) = f 3 0 = 3 - 2·0 = 3

e = 0

Thus it evaluates to 9