U23SA861Y : or have it be accumulateWithStride and have the 16 come in as a parameter
U23SA861Y : The only thing not specified would be what happens to the empty list
U48AEBJQ3 : I think I would like `sumDigits : Int -&gt; List Int -&gt; Int` better.
U48AEBJQ3 : But names are hard.
U29JSAR9S : just trying to test it out as I'm not convinced it does actually solve it (I think it might miss the case when you don't multiply by a power at all for the first hex digit) - getting:

```
the argument to function `length` is causing a mismatch.

120|            List.length mapped
                     ^^^^^^
Function `length` is expecting the argument to be:

    List b

But it is:

    List (Maybe b) -&gt; List b
```

for the maybeList function - and I'm too tired to figure out whats up myself :slightly_smiling_face:

U23SA861Y : ahh the line should be `List.filterMap identity l`
U23SA861Y : my bad
U23SA861Y : well the first digit should be multplied by one, the second by 16 and so on
U29JSAR9S : ah, of course
U23SA861Y : the one thing it does to would be return 0 for an empty list as opposed to say Nothing
U48AEBJQ3 : How about

```
maybeList : List (Maybe a) -&gt; Maybe (List a)
maybeList =
    List.foldr (Maybe.map2 (::)) (Just [])
```

U23SA861Y : hmm, yes I think that would do what you want
U29JSAR9S : cheers guys, been interesting working through this with you - my understanding of how to handle Maybe's has definitely come along this evening!
U23SA861Y : no problem, and good luck
U23SA861Y : in your further learnins for great profit
U29JSAR9S : last one for you, for the road:

```
maybeTuple3 : ( Maybe a, Maybe b, Maybe c ) -&gt; Maybe ( a, b, c )
maybeTuple3 tuple =
    case tuple of
        ( Just a, Just b, Just c ) -&gt;
            Just ( a, b, c )

        _ -&gt;
            Nothing
```

Trying to figure out how to do this with a Maybe.mapX

U23SA861Y : I think you probably want curry3 or something like that
U23SA861Y : mmm there is only curry and uncurry no uncurry 3
U23SA861Y : but perhaps you could write it

```
uncurry3 : (a -&gt; b -&gt; c -&gt; d) -&gt; (a,b,c) -&gt; d
uncurry3 f (a,b,c) = f a b c
```

U23SA861Y : then it would go together something like `maybeTuple3 = uncurry3 (Maybe.map3 (,,))`
U23SA861Y : I think, this is me coding without checking if it compiles

U29JSAR9S : this works :```

```
maybeTuple3 : ( Maybe a, Maybe b, Maybe c ) -&gt; Maybe ( a, b, c )
maybeTuple3 ( a, b, c ) =
    Maybe.map3 (,,) a b c
```

U29JSAR9S : I didn't realise you could do the (,,) thing
U23SA861Y : well there you go
U29JSAR9S : cheers :slightly_smiling_face: