```
[2, 1, 7, 6, 6, 6, 6, 6, 9, 6, 5, 1, 7, 5, 8, 8, 3, 3, 9, 1,    more ,6, 6, 5, 8, 4, 7, 4, 2,
```

```
begin
    # Submarine movements
    movements = [split(el, " ") for el in readlines("sub_movement.txt")]
    direction = getindex.(movements,1)
    # Use "dot call" syntax
    amounts = parse.(Int, getindex.(movements,2))
end
```

```
position (generic function with 1 method)
```

```
# functions - https://docs.julialang.org/en/v1/manual/functions/#man-functions
# short-circuit evaluation (&&) -
https://docs.julialang.org/en/v1/manual/functions/#man-functions
function position(direction, amount)
    horizontal = 0
    vertical = 0

    for (direction, amount) in zip(direction, amount)
        # In the expression a && b, the subexpression b is
        # only evaluated if a evaluates to true
        direction == "forward" && (horizontal += amount)
        direction == "down" && (vertical += amount)
        direction == "up" && (vertical -= amount)
    end

    return (horizontal, vertical)
end
```

```
1694130
```

```
prod(position(direction, amounts))
```

```
position_aim_adj (generic function with 1 method)
```

```
function position_aim_adj(direction, amount)
    horizontal = 0
    vertical = 0
    aim = 0

    for (direction, amount) in zip(direction, amount)
        if direction == "forward"
            horizontal += amount
            vertical = vertical + (aim * amount)
        end
        direction == "down" && (aim += amount)
        direction == "up" && (aim -= amount)
    end

    return (horizontal, vertical)
end
```

```
1698850445
```

```
prod(position_aim_adj(direction, amounts))
```

Cell deleted (**UNDO**)