```
fun nums-to-points(measure :: Number, platform :: String):
  ...
where:
  nums-to-points(15, "facebook") is 15 * 1
  nums-to-points(15, "snapchat") is 15 * 1
  nums-to-points(15, "news") is 15 * 3
end

fun points-to-rating(points :: Number, location :: String):
  ...
where:
  points-to-rating(121, "") is "serious"
  points-to-rating(100, "school") is "moderate"
  points-to-rating(70, "work") is "moderate"
  points-to-rating(70, "home") is "normal"
  points-to-rating(9, "") is "mild"
  points-to-rating(40, "work") is "normal"
end

fun binge-rating1(
  fbmin :: Number, scmin :: Number,
  news-checks :: Number, location :: String) -> String:
  total-points =
     nums-to-points(fbmin, "facebook") +
     nums-to-points(scmin, "snapchat") +
     nums-to-points(news-checks, "news")
  points-to-rating(total-points, location)
end
```

```
fun soc-media-points(mins :: Number):
  doc: "multiply given social-media minutes by 1"
  ...
end

fun news-points(checks :: Number):
  doc: "multiply given social-media minutes by 3"
  ...
end

fun binge-rating2(
    fbmin :: Number, scmin :: Number,
    news-checks :: Number, location :: String) -> String:
  fb-points = soc-media-points(fbmin)
  sc-points = soc-media-points(scmin)
  news-points = news-points(news-checks)
  total-points = fb-points + sc-points + news-points
  points-to-rating(total-points, location)
end
```

```
fun usage-to-points(usage :: Number, factor :: Number):
  ...
end

fun binge-rating3(
    fbmin :: Number, scmin :: Number,
    news-checks :: Number, location :: String) -> String:
  total-points =
    usage-to-points(fbmin, 1) +
    usage-to-points(scmin, 1) +
    usage-to-points(news-checks, 3)
  points-to-rating(total-points, location)
end
```