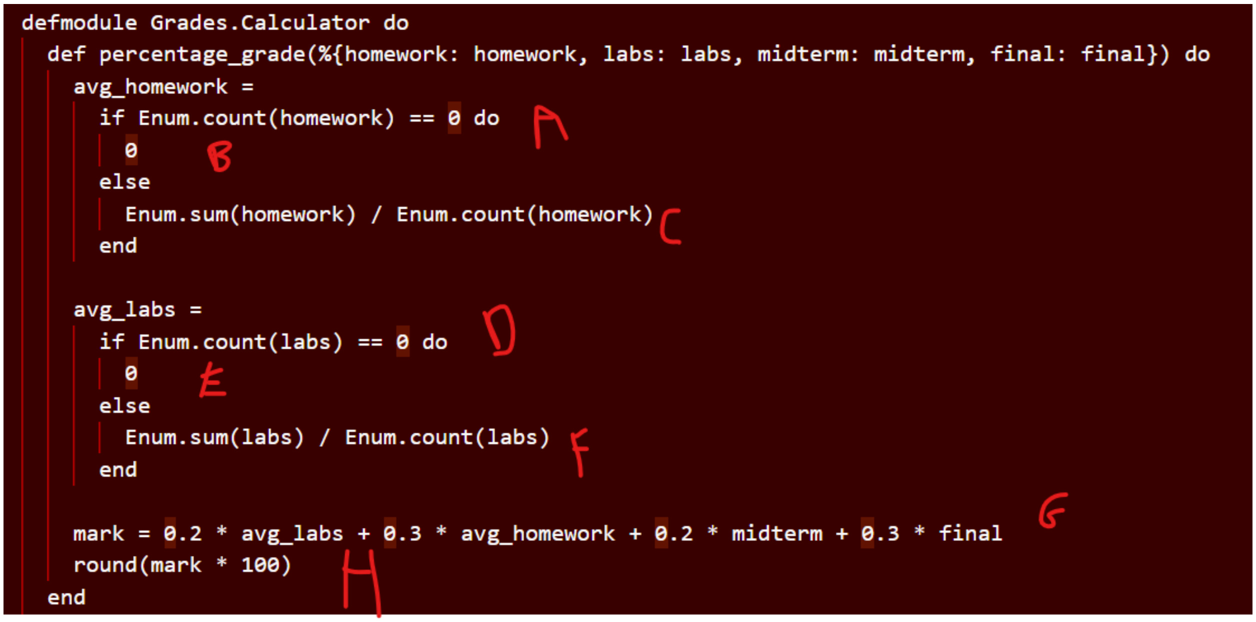
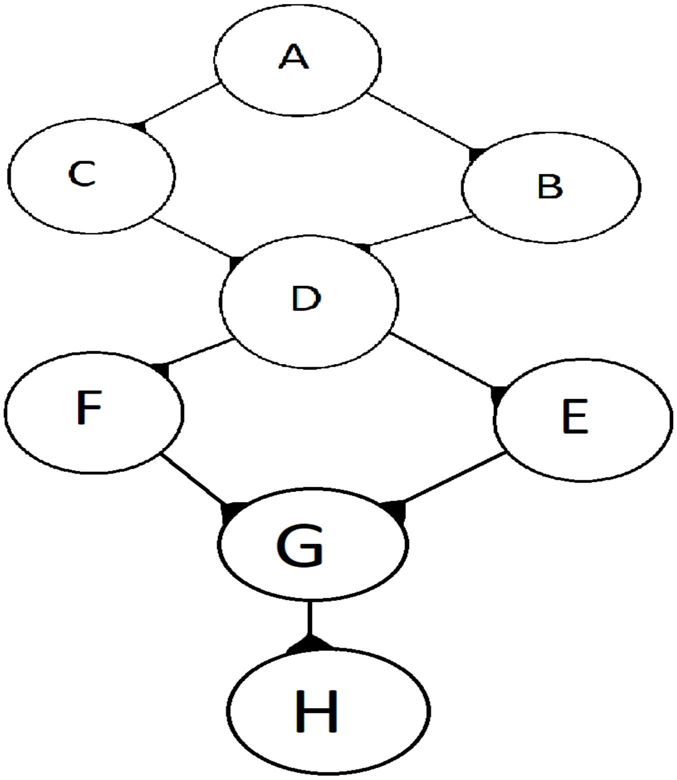
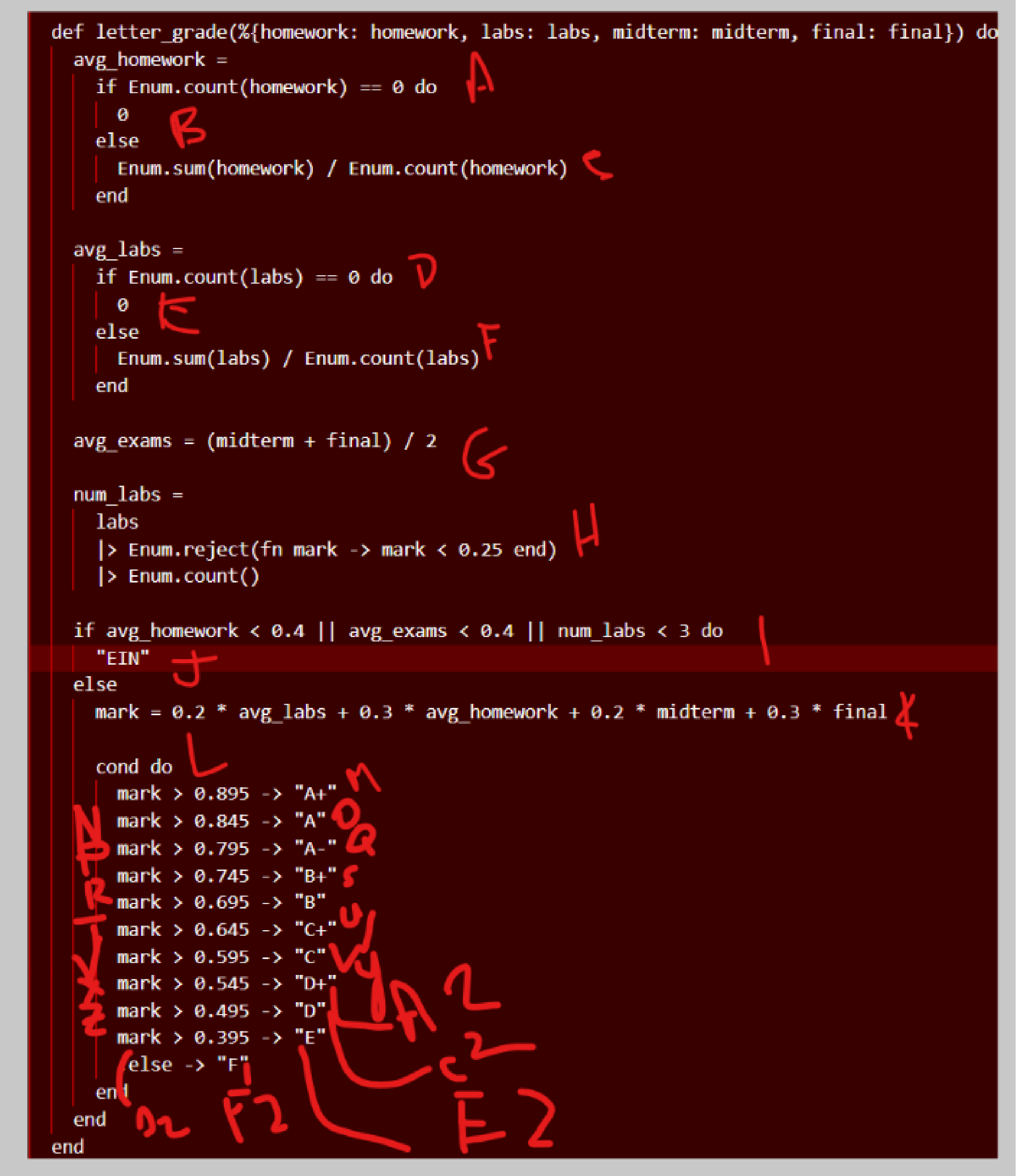
Problem 1.1

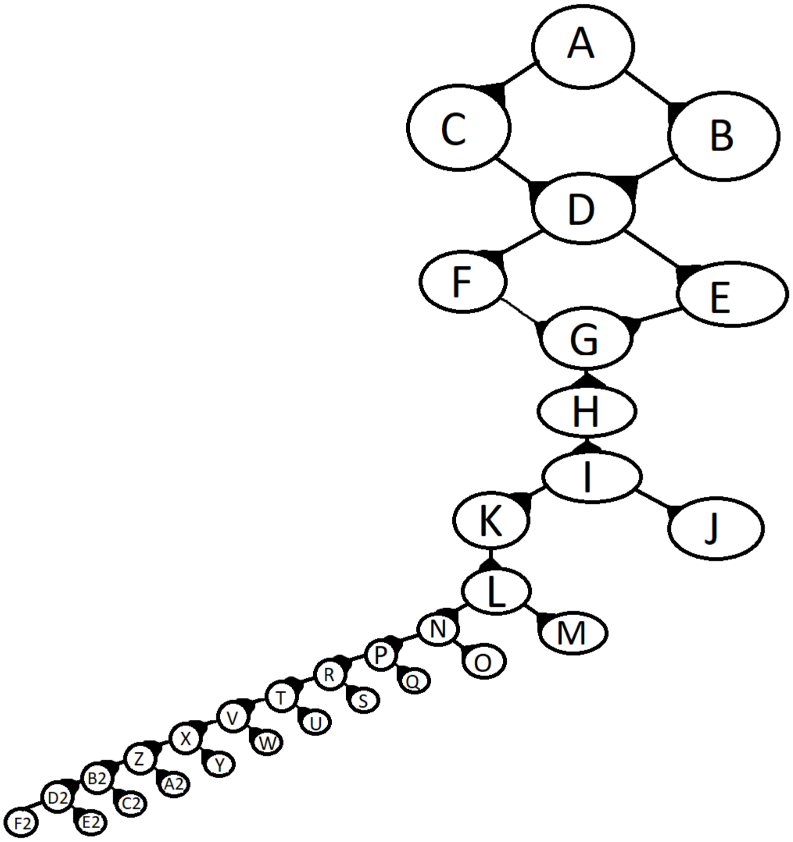


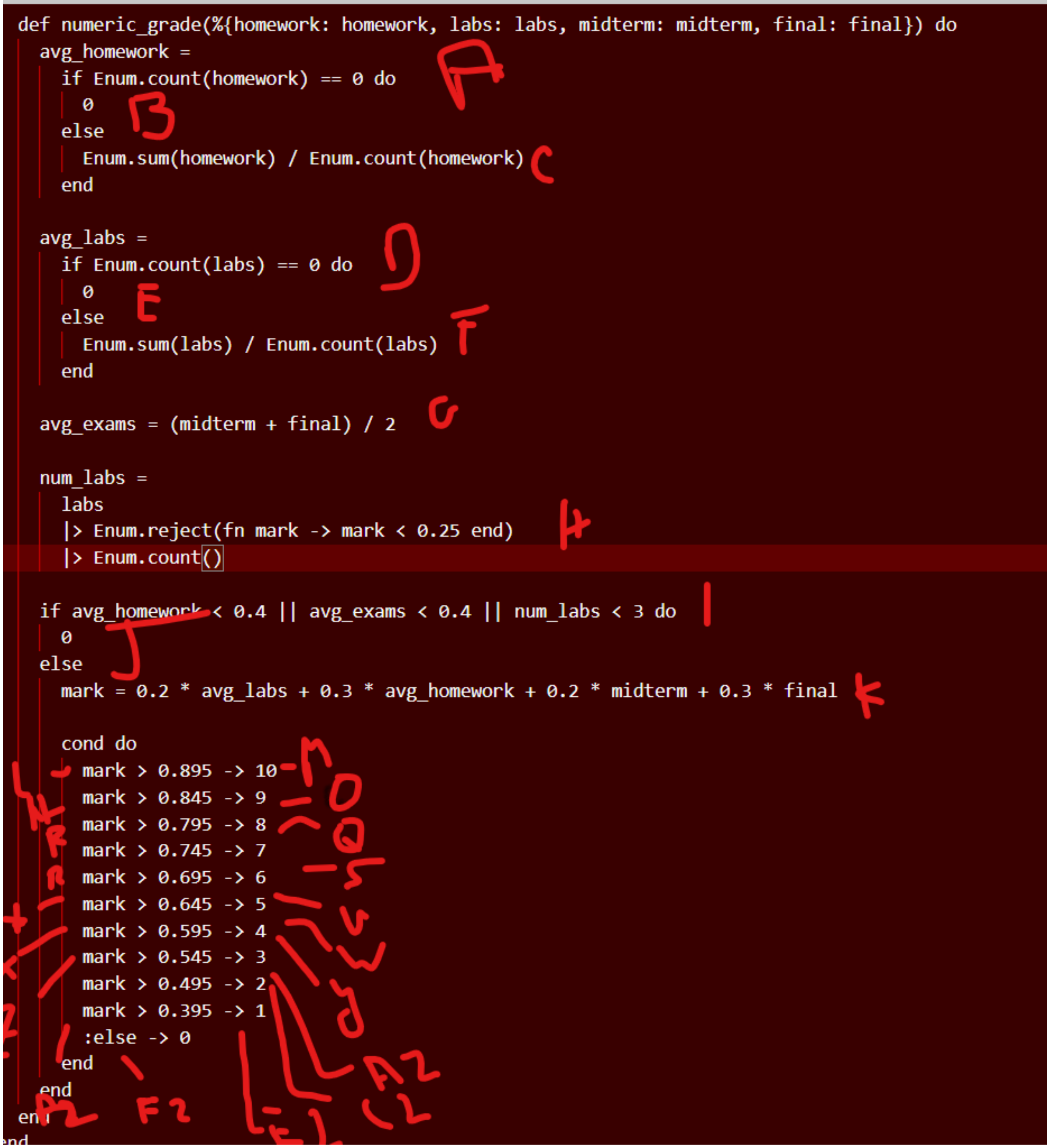
```
defmodule Grades.Calculator do
  def percentage_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
    avg_homework =
      if Enum.count(homework) == 0 do
        0
      else
        Enum.sum(homework) / Enum.count(homework)
      end

    avg_labs =
      if Enum.count(labs) == 0 do
        0
      else
        Enum.sum(labs) / Enum.count(labs)
      end

    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
    round(mark * 100)
  end
```



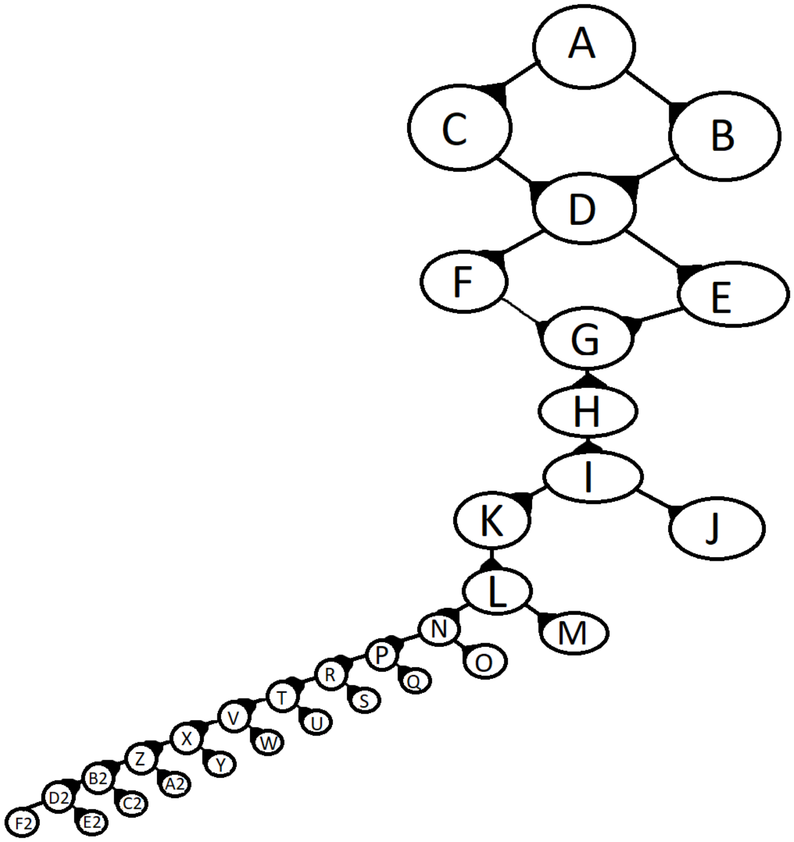


```
def letter_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
  avg_homework =
    if Enum.count(homework) == 0 do
      0
    else
      Enum.sum(homework) / Enum.count(homework)
    end

  avg_labs =
    if Enum.count(labs) == 0 do
      0
    else
      Enum.sum(labs) / Enum.count(labs)
    end

  avg_exams = (midterm + final) / 2

  num_labs =
    labs
    |> Enum.reject(fn mark -> mark < 0.25 end)
    |> Enum.count()

  if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
    "EIN"
  else
    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final

    cond do
      mark > 0.895 -> "A+"
      mark > 0.845 -> "A"
      mark > 0.795 -> "A-"
      mark > 0.745 -> "B+"
      mark > 0.695 -> "B"
      mark > 0.645 -> "C+"
      mark > 0.595 -> "C"
      mark > 0.545 -> "D+"
      mark > 0.495 -> "D"
      mark > 0.395 -> "E"
      else -> "F"
    end
  end
end
```


A
C
B
D
F
E
G
H
I
K
J
L
N
M
O
P
Q
R
S
T
U
V
W
X
Y
Z
A2
B2
C2
D2
E2
F2



```
def numeric_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
  avg_homework =
    if Enum.count(homework) == 0 do
      0
    else
      Enum.sum(homework) / Enum.count(homework)
    end

  avg_labs =
    if Enum.count(labs) == 0 do
      0
    else
      Enum.sum(labs) / Enum.count(labs)
    end

  avg_exams = (midterm + final) / 2

  num_labs =
    labs
    |> Enum.reject(fn mark -> mark < 0.25 end)
    |> Enum.count()

  if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
    0
  else
    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final

    cond do
      mark > 0.895 -> 10
      mark > 0.845 -> 9
      mark > 0.795 -> 8
      mark > 0.745 -> 7
      mark > 0.695 -> 6
      mark > 0.645 -> 5
      mark > 0.595 -> 4
      mark > 0.545 -> 3
      mark > 0.495 -> 2
      mark > 0.395 -> 1
      :else -> 0
    end
  end
end
end
```


A
C
B
D
F
E
G
H
I
K
J
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
A2
B2
C2
D2
E2
F2

Problem 1.2

Test Case Number	Test Data	Expected results	Conditions Covered	Branches Covered
1	Calculator.percentage_grade (%{ homework: [], labs: [], midterm: 0.70, final: 0.9 })	41	Enum.count(homework) == 0 Enum.count(labs) == 0	2/4 or 50%
2	Calculator.percentage_grade (%{ homework: [0.8], labs: [1, 1, 1], midterm: 0.70, final: 1 })	88	Enum.count(homework) != 0 Enum.count(labs) != 0	4/4 100%
3	Calculator.letter_grade(%{ homework: [], labs: [], midterm: 0.70, final: 1 })	“EIN”	Enum.count(homework) == 0 Enum.count(labs) == 0 avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3	3/16
4	Calculator.letter_grade(%{ homework: [1], labs: [1,1,1], midterm: 1,	“A+”	Enum.count(homework) != 0 Enum.count(labs) != 0	6/16

	final: 1 })		!(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.895	
5	Calculator.letter_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0.5 })	“A”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.845	7/16
6	Calculator.letter_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0.333 })	“A-”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.795	8/16
7	Calculator.letter_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0.17	“B+”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams <	9/16

	})		0.4 || num_labs < 3) mark > 0.745	
8	Calculator.letter_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0 })	“B”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.695	10/16
9	Calculator.letter_grade(%{ homework: [0.84], labs: [1,1,1], midterm: 1, final: 0 })	“C+”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.645	11/16
10	Calculator.letter_grade(%{ homework: [0.67], labs: [1,1,1], midterm: 1, final: 0 })	“C”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3)	12/16

			mark > 0.595	
11	Calculator.letter_grade(%{ homework: [0.5], labs: [1,1,1], midterm: 1, final: 0 })	"D+"	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.545	13/16
12	Calculator.letter_grade(%{ homework: [0.5], labs: [0.75,0.75,0.75], midterm: 1, final: 0 })	"D"	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.495	14/16
13	Calculator.letter_grade(%{ homework: [0.5], labs: [0.5,0.5,0.5], midterm: 1, final: 0 })	"E"	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.395	15/16

14	Calculator.letter_grade(%{ homework: [0.4], labs: [0.4,0.4,0.4], midterm: 0.8, final: 0 })	“F”	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark <0.395 (else)	16/16
15	numeric_grade(%{ homework: [], labs: [], midterm: 0.70, final: 1 })	0	Enum.count(homework) == 0 Enum.count(labs) == 0 avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3	3/16
16	numeric_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 1 })	10	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.895	6/16
17	numeric_grade(%{ homework:	9	Enum.count(homework) != 0	7/16

	[1], labs: [1,1,1], midterm: 1, final: 0.5 })		Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.845	
18	numeric_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0.333 })	8	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.795	8/16
19	numeric_grade(%{ homework: [1], labs: [1,1,1], midterm: 1, final: 0.17 })	7	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.745	9/16
20	numeric_grade(%{ homework: [1], labs: [1,1,1], midterm: 1,	6	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homewo	10/16

	final: 0 })		rk < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.695	
21	numeric_grade(%{ homework: [0.84], labs: [1,1,1], midterm: 1, final: 0 })	5	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.645	11/16
22	numeric_grade(%{ homework: [0.67], labs: [1,1,1], midterm: 1, final: 0 })	4	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.595	12/16
23	numeric_grade(%{ homework: [0.5], labs: [1,1,1], midterm: 1, final: 0 })	3	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs	13/16

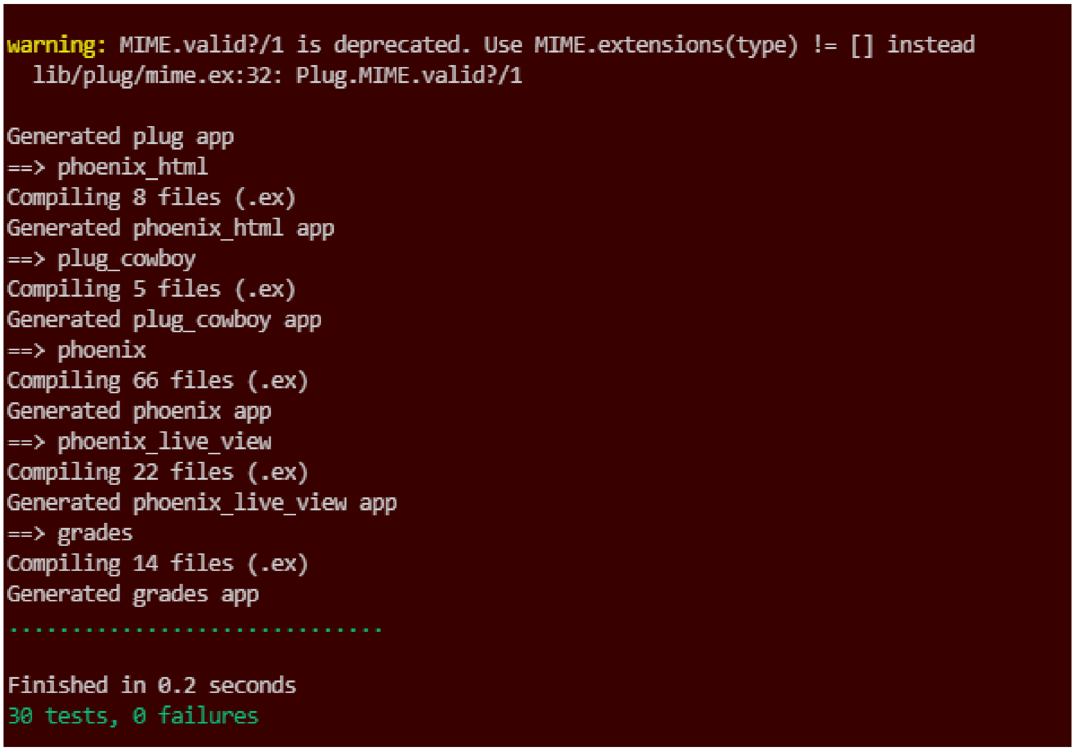
			< 3) mark > 0.545	
24	numeric_grade(%{ homework: [0.5], labs: [0.75,0.75,0.75], midterm: 1, final: 0 })	2	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.495	14/16
25	numeric_grade(%{ homework: [0.5], labs: [0.5,0.5,0.5], midterm: 1, final: 0 })	1	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3) mark > 0.395	15/16
26	numeric_grade(%{ homework: [0.4], labs: [0.4,0.4,0.4], midterm: 0.8, final: 0 })	0	Enum.count(homework) != 0 Enum.count(labs) != 0 !(avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3)	16/16

			mark <0.395 (else)	

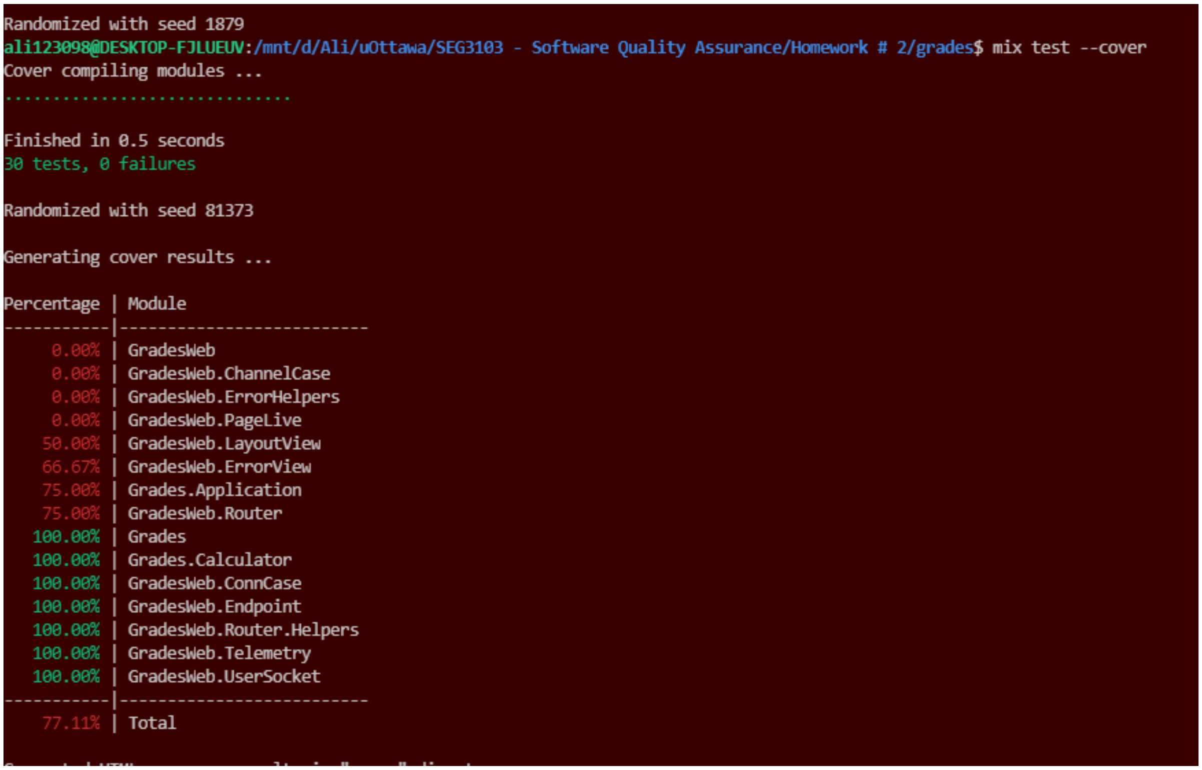
Problem 1.3

https://github.com/Renfrew100/seg3103_playground/blob/master/Homework%20%23%202/grades/test/grades/calculator_test.exs

Problem 1.4



```
warning: MIME.valid?/1 is deprecated. Use MIME.extensions(type) != [] instead
  lib/plug/mime.ex:32: Plug.MIME.valid?/1

Generated plug app
==> phoenix_html
Compiling 8 files (.ex)
Generated phoenix_html app
==> plug_cowboy
Compiling 5 files (.ex)
Generated plug_cowboy app
==> phoenix
Compiling 66 files (.ex)
Generated phoenix app
==> phoenix_live_view
Compiling 22 files (.ex)
Generated phoenix_live_view app
==> grades
Compiling 14 files (.ex)
Generated grades app
..............................

Finished in 0.2 seconds
30 tests, 0 failures
```



```
Randomized with seed 1879
ali123098@DESKTOP-FJLUEUV:/mnt/d/Ali/uOttawa/SEG3103 - Software Quality Assurance/Homework # 2/grades$ mix test --cover
Cover compiling modules ...
..............................

Finished in 0.5 seconds
30 tests, 0 failures

Randomized with seed 81373

Generating cover results ...

Percentage | Module
-----------|--------------------------
     0.00% | GradesWeb
     0.00% | GradesWeb.ChannelCase
     0.00% | GradesWeb.ErrorHelpers
     0.00% | GradesWeb.PageLive
    50.00% | GradesWeb.LayoutView
    66.67% | GradesWeb.ErrorView
    75.00% | Grades.Application
    75.00% | GradesWeb.Router
   100.00% | Grades
   100.00% | Grades.Calculator
   100.00% | GradesWeb.ConnCase
   100.00% | GradesWeb.Endpoint
   100.00% | GradesWeb.Router.Helpers
   100.00% | GradesWeb.Telemetry
   100.00% | GradesWeb.UserSocket
-----------|--------------------------
    77.11% | Total
```

Problem 2 (2.1 - 2.4):

Refer to the file for helpful functions and look at commit history :
https://github.com/Renfrew100/seg3103_playground/blob/master/Homework%20%23%202/grades/lib/grades/calculator.ex