SEG3103 Assignment 2

Haochu Chen

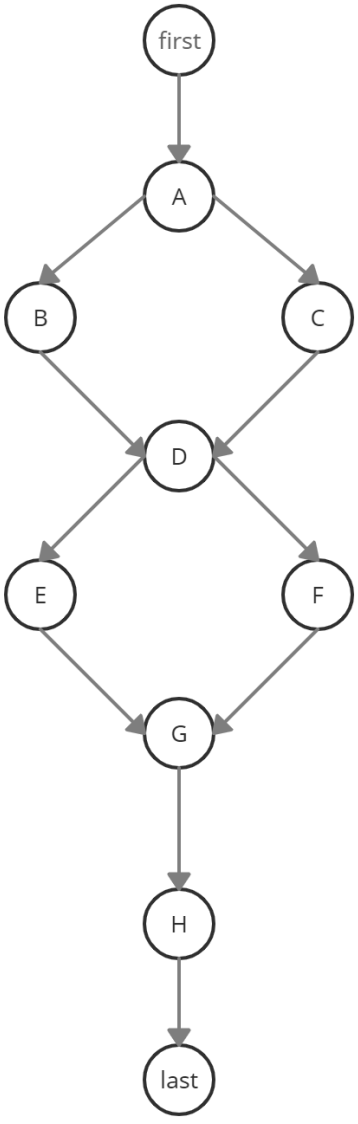
300067058

Problem1

1.1

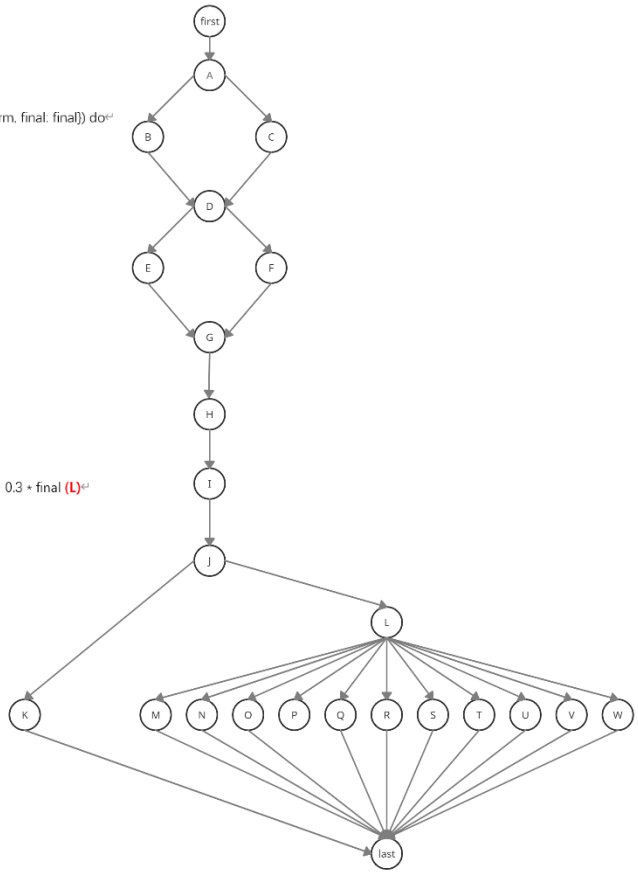
percentage_grade

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

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

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



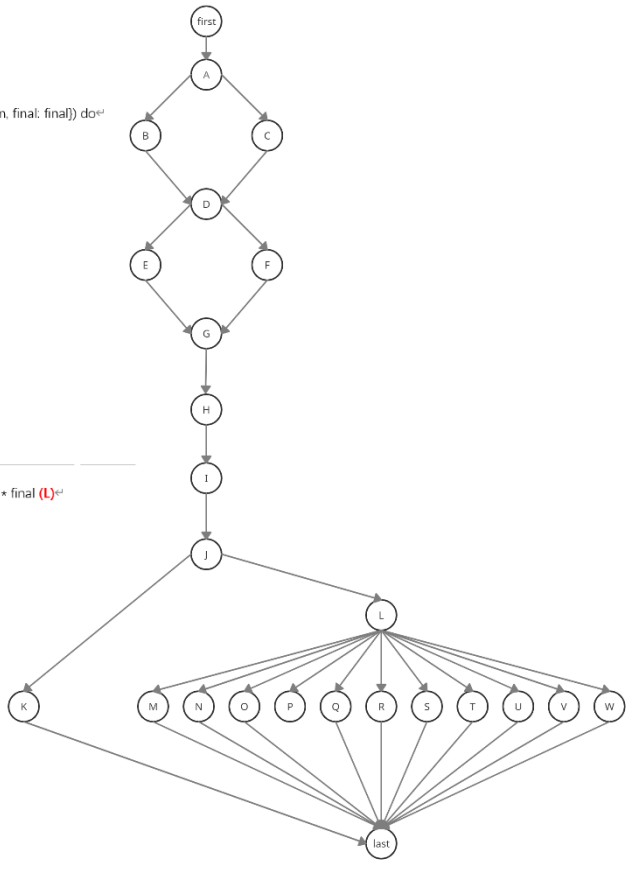
letter_grade

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

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

  avg_exams = (midterm + final) / 2 (G)

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

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

    cond do
      mark > 0.895 -> "A+" (M)
      mark > 0.845 -> "A" (N)
      mark > 0.795 -> "A-" (O)
      mark > 0.745 -> "B+" (P)
      mark > 0.695 -> "B" (Q)
      mark > 0.645 -> "C+" (R)
      mark > 0.595 -> "C" (S)
      mark > 0.545 -> "D+" (T)
      mark > 0.495 -> "D" (U)
      mark > 0.395 -> "E" (V)
      :else -> "F" (W)
    end
  end
end
```



numeric_grade

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

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

  avg_exams = (midterm + final) / 2 (G)

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

  if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do (J)
    0 (K)

  else
    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final (L)

    cond do
      mark > 0.895 -> 10 (M)
      mark > 0.845 -> 9 (N)
      mark > 0.795 -> 8 (O)
      mark > 0.745 -> 7 (P)
      mark > 0.695 -> 6 (Q)
      mark > 0.645 -> 5 (R)
      mark > 0.595 -> 4 (S)
      mark > 0.545 -> 3 (T)
      mark > 0.495 -> 2 (U)
      mark > 0.395 -> 1 (V)
      :else -> 0 (W)
    end
  end
end
```



1.2

percentage_grade:

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

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

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

Test Case Number	Test Data	Expected Results	Conditions Covered	Branches Covered
1	homework: [1], labs: [0], midterm: 1, final: 1	80	AD	CEGH
2	homework: [0], labs: [1, 1, 1], midterm: 1, final: 1	70	AD	BFGH

letter_grade:

See the next page

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

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

  avg_exams = (midterm + final) / 2 (G)

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

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

    cond do
      mark > 0.895 -> "A+" (M)
      mark > 0.845 -> "A" (N)
      mark > 0.795 -> "A-" (O)
      mark > 0.745 -> "B+" (P)
      mark > 0.695 -> "B" (Q)
      mark > 0.645 -> "C+" (R)
      mark > 0.595 -> "C" (S)
      mark > 0.545 -> "D+" (T)
      mark > 0.495 -> "D" (U)
      mark > 0.395 -> "E" (V)
      :else -> "F" (W)
    end
  end
end
```

Test Case Number	Test Data	Expected Results	Conditions Covered	Branches Covered
1	homework: [0], labs: [0], midterm: 0, final: 0	EIN	ADJW	BEGHIJKW
2	homework: [1], labs: [1, 1, 1], midterm: 1, final: 1	A+	ADJM	CFGHIJLM
3	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.5	A	ADJN	CFGHIJLN
4	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.35	A-	ADJO	CFGHIJLO

5	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.2	B+	ADJP	CFGHIJLP
6	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.1	B	ADJQ	CFGHIJLQ
7	homework: [1], labs: [1, 1, 1], midterm: 0.9, final: 0	C+	ADJR	CFGHIJLR
8	homework: [0.7], labs: [1, 1, 1], midterm: 0.4, final: 0.4	C	ADJS	CFGHIJLS
9	homework: [0.6], labs: [1, 1, 1], midterm: 0.4, final: 0.4	D+	ADJT	CFGHIJLT
10	homework: [0.4], labs: [1, 1, 1], midterm: 0.4, final: 0.4	D	ADJU	CFGHIJLU
11	homework: [0.4], labs: [0.4, 0.4, 0.4], midterm: 0.4, final: 0.4	E	ADJV	CFGHIJLV
12	Cannot be reached	F	ADJW	CFGHIJLW

numeric_grade:

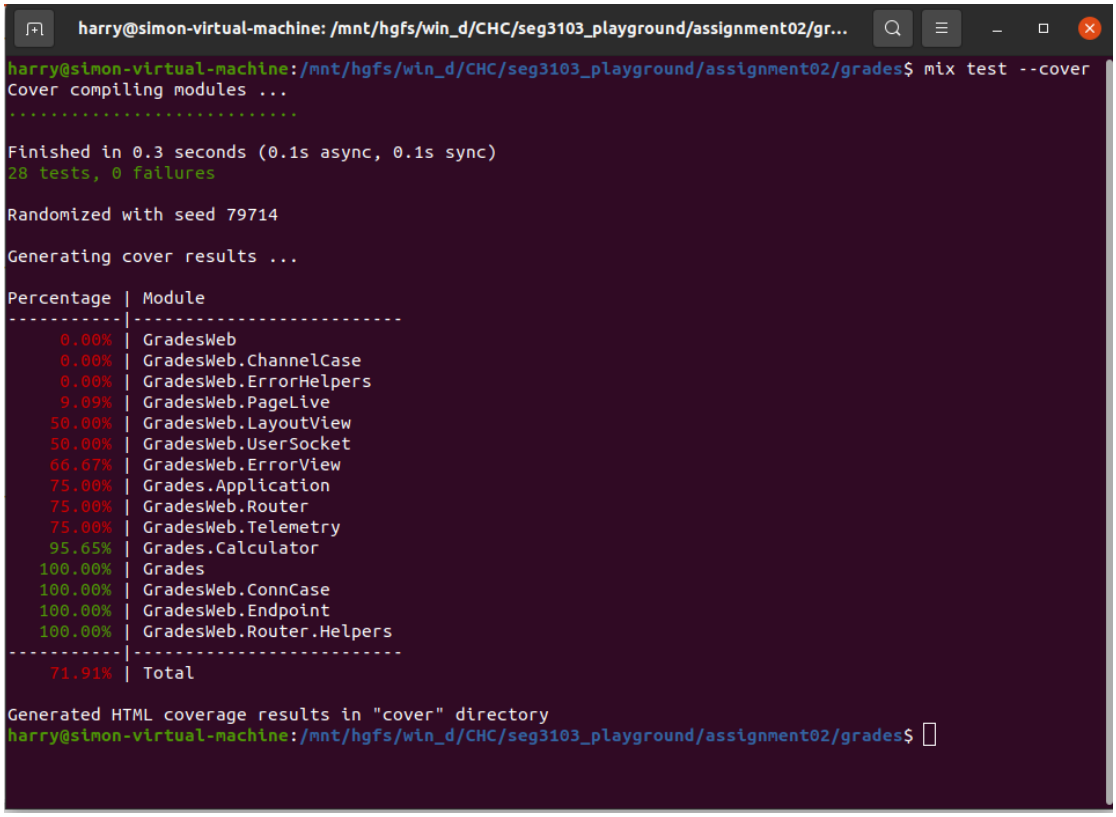
See the next page

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

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

  avg_exams = (midterm + final) / 2 (G)

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

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

    cond do
      mark > 0.895 -> 10 (M)
      mark > 0.845 -> 9 (N)
      mark > 0.795 -> 8 (O)
      mark > 0.745 -> 7 (P)
      mark > 0.695 -> 6 (Q)
      mark > 0.645 -> 5 (R)
      mark > 0.595 -> 4 (S)
      mark > 0.545 -> 3 (T)
      mark > 0.495 -> 2 (U)
      mark > 0.395 -> 1 (V)
      :else -> 0 (W)
    end
  end
end
end
```

Test Case Number	Test Data	Expected Results	Conditions Covered	Branches Covered
1	homework: [0], labs: [0], midterm: 0, final: 0	0	ADJW	BEGHIJKW
2	homework: [1], labs: [1, 1, 1], midterm: 1, final: 1	10	ADJM	CFGHIJLM
3	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.5	9	ADJN	CFGHIJLN
4	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.35	8	ADJO	CFGHIJLO
5	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.2	7	ADJP	CFGHIJLP

6	homework: [1], labs: [1, 1, 1], midterm: 1, final: 0.1	6	ADJQ	CFGHIJLQ
7	homework: [1], labs: [1, 1, 1], midterm: 0.9, final: 0	5	ADJR	CFGHIJLR
8	homework: [0.7], labs: [1, 1, 1], midterm: 0.4, final: 0.4	4	ADJS	CFGHIJLS
9	homework: [0.6], labs: [1, 1, 1], midterm: 0.4, final: 0.4	3	ADJT	CFGHIJLT
10	homework: [0.4], labs: [1, 1, 1], midterm: 0.4, final: 0.4	2	ADJU	CFGHIJLU
11	homework: [0.4], labs: [0.4, 0.4, 0.4], midterm: 0.4, final: 0.4	1	ADJV	CFGHIJLV

1.4



The coverage is 95.65%, I am not able to achieve 100% coverage, because for letter_grade and numeric_grade, If I want to check the ":else -> "F" " and ":else -> 0" branches, then homework, labs, midterm, and final, at least one of them must be less than 0.395, and the remaining three must be equal to 0.395. And this will cause the program go to the K branch and return the "EIN" or 0 directly then stop the program, and won't check the conditions and branches from M to W.

```
if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do (J)
   "EIN" (K)
else
   mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final (L)

   cond do
      mark > 0.895 -> "A+" (M)
      mark > 0.845 -> "A" (N)
      mark > 0.795 -> "A-" (O)
      mark > 0.745 -> "B+" (P)
      mark > 0.695 -> "B" (Q)
      mark > 0.645 -> "C+" (R)
      mark > 0.595 -> "C" (S)
      mark > 0.545 -> "D+" (T)
      mark > 0.495 -> "D" (U)
      mark > 0.395 -> "E" (V)
      :else -> "F" (W)
```

```
if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do (J)
   0 (K)
else
   mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final (L)

   cond do
      mark > 0.895 -> 10 (M)
      mark > 0.845 -> 9 (N)
      mark > 0.795 -> 8 (O)
      mark > 0.745 -> 7 (P)
      mark > 0.695 -> 6 (Q)
      mark > 0.645 -> 5 (R)
      mark > 0.595 -> 4 (S)
      mark > 0.545 -> 3 (T)
      mark > 0.495 -> 2 (U)
      mark > 0.395 -> 1 (V)
      :else -> 0 (W)
```

Problem2:

Attached in the zip file or you can check it in my github repository:
https://github.com/Harrry1314/seg3103_playground.git

Thanks for your time.