hgg26@hi.is
rao13@hi.is

Assignment 8

1. **Derive test cases (i.e. input and output values) using equivalence class partitioning and choose representative values using boundary value analysis (only boundary values, but no typical usage values apply). As coverage criterion, aim for 100% coverage of equivalence classes of valid and invalid input values and 100% coverage of boundary values. In addition, aim also for a 100% coverage equivalence of classes and boundary values of output**

Valid input

Kennitala: 101000000, 3112999999

Programme: 1,2

Grade: 0.0, 10

Valid Equivalent Classes

Kt[101000000, 3112999999]	ValidKennitalEquivalentClass1
Pro[1,2]	ValidProgrammeEquivalentClass1
Grade[9.0,10,0]	ValidGradeEquivalentClass1
Grade[7.25,9.0)	ValidGradeEquivalentClass2
Grade[6.0,7,25)	ValidGradeEquivalentClass3
Grade[0.0,6.0)	ValidGradeEquivalentClass4

Invalid Equivalent Classes

Kt < 101000000	InvalidKennitalEquivalentClass1
Kt > 3112999999	InvalidKennitalEquivalentClass1
Pro != 1, !=2	InvalidProgrammeEquivalentClass1
Grade <0	InvalidGradeEquivalentClass1
Grade>10	InvalidGradeEquivalentClass2

hgg26@hi.is
rao13@hi.is

Boundary Values

Kennitala [100999999,101000000,3112999999,3113000000]

Programme [1,2,3,4]

Grade [-0,01| 0.0 | 5,9 |6,1 |7,24 |7,25 |8,9 |9,0 |10.0 | 10,0000001]

Test Cases

PrintGrades(100999999,2,5.9) →	ERROR
printGrades(100999999,3,10.0) →	ERROR
printGrades(100999999,1,10.0) →	ERROR
printGrades(100999999,2,10.0) →	ERROR
PrintGrades(101000000,2,5.6) →	‘FAIL‘
printGrades(101000000,2,9.0) →	“HONORS“
printGrades(101000000,2,10,0000001) →	“HONORS“
printGrades(101000000,2,10,000001) →	ERROR
PrintGrades(3112999999,1,10.0) →	“HONORS“
printGrades(3112999999,3,0.0) →	ERROR
printGrades(3112999999,1,7.24) →	“Second Class“
printGrades(3112999999,2,10.0) →	“HONORS“
printGrades(3113000000,2,8.9) →	ERROR
printGrades(3113000000,1,7.24) →	ERROR
printGrades(3113000000,2,7.24) →	ERROR
printGrades(3113000000,4,7.24) →	ERROR