Black Box Test Plan

```
Files:
input/
     smallmap.txt
          2 0 7.0 159.0
          3 2 12.0 212.0
          0 3 14.0 415.0
          1 0 5.0 99.0
          3 1 10.0 112.0
          1 2 6.0 72.0

     invalidmap.txt
          2 -0 7.0 159.0
          3 2 12.0 212.0
          0 3 14.0 -415.0
          -1 0 5.0 99.0
          3 1 -10.0 112.0
          1 2 6.0 72.0

     invalidboundarymap.txt
          0 0 7.0 159.0
          0 0 12.0 212.0
          0 3 14.0 415.0
          1 0 5.0 99.0
          3 1 10.0 112.0
          1 2 6.0 72.0

     illegalcharacters.txt
          #($)!^?:;
```

Test ID	Description	Expected Results	Actual Results

ECP - Valid file entered to list	Run the program with no extra arguments.
	-Program asks for a map:
	input/smallmap.txt
	-Program asks to minimize cost or asphalt:
	cost
		ConsoleOut:
		Heap[
		Highway[city1=X, city2=X, cost=X.X, asphalt=X.X],
		Highway[city1=X, city2=X, cost=X.X, asphalt=X.X],
		Highway[city1=X, city2=X, cost=X.X, asphalt=X.X]
		]
		List[
		Highway[city1=X, city2=X, cost=X.X, asphalt=X.X],
		Highway[city1=X, city2=X, cost=X.X, asphalt=X.X],
		Highway[city1=X, city2=X, cost=X.X,
			Heap[
			Highway[city1=1, city2=0, cost=5.0, asphalt=99.0],
			Highway[city1=2, city2=0, cost=7.0, asphalt=159.0],
			Highway[city1=1, city2=2, cost=6.0, asphalt=72.0],
			Highway[city1=3, city2=2, cost=12.0, asphalt=212.0],
			Highway[city1=3, city2=1, cost=10.0,

		asphalt=X.X]
		]
		AdjacencyList[
		City 0: -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X] -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X] -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X]
		City 1: -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X] -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X]
		City 2: -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X] -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X]
			asphalt=112.0],
			Highway[city1=0, city2=3, cost=14.0, asphalt=415.0]
			]
			List[
			Highway[city1=1, city2=0, cost=5.0, asphalt=99.0],
			Highway[city1=1, city2=2, cost=6.0, asphalt=72.0],
			Highway[city1=3, city2=1, cost=10.0, asphalt=112.0]
			]
			AdjacencyList[
			City 0: -> Highway[city

		-> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X] City 3: -> Highway[city1=X, city2=X, cost=X.X, asphalt=X.X]]	1=0, city2=3, cost=14.0, asphalt=415.0] -> Highway[city1=1, city2=0, cost=5.0, asphalt=99.0] -> Highway[city1=2, city2=0, cost=7.0, asphalt=159.0] City 1: -> Highway[city1=1, city2=0, cost=5.0, asphalt=99.0] -> Highway[city1=1, city2=2, cost=6.0, asphalt=72.0] -> Highway[city1=3, city2=1, cost=10.0, asphalt=112.0]

			City 2: -> Highway[city1=1, city2=2, cost=6.0, asphalt=72.0] -> Highway[city1=2, city2=0, cost=7.0, asphalt=159.0] -> Highway[city1=3, city2=2, cost=12.0, asphalt=212.0] City 3: -> Highway[city1=0, city2=3, cost=14.0, asphalt=415.0] -> Highway[city1=3, city2=1, cost=10.0, asphalt=112.0] -> Highway[city1=3, city2=2, cost=12.0,

			asphalt=212.0]]
ECP - Invalid word checked	Run the program with no extra arguments. -Program asks for highway map: input/invalidmap.txt	ConsoleOut: Invalid argument: negative number	Error: Invalid file! An IllegalArgumentException is thrown
BVA - Check word with multiple spell checking rules	Run the program without any extra arguments. -Program asks for highway map: input/invalidboundarymap.txt	ConsoleOut: Invalid Argument: Road from and to same city Or Invalid Argument: Two different entries for same road	Error: Road from and to same city! An IllegalArgumentException is thrown
DT - Illegal Characters in text file	Program is run without any extra arguments. -Program asks for highway map: input/illegalcharacters.txt	Program stops after encountering error ‘IllegalArgument’, exits with the following output	Error: Invalid file! An IllegalArgumentException is thrown

		ConsoleOut:Invalid Argument: Invalid characters	