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 -	Run the program		Неар[
Valid	with no extra	ConsoleOut:	
file	arguments.	Heap[Highway[city
entered			1=1,
to list	-Program asks for a	Highway[city1=	city2=0,
	map:	X, city2=X,	cost=5.0,
	input/smallmap.txt	cost=X.X,	asphalt=99.0
	-Program asks to	asphalt=X.X],],
	minimize cost or		
	asphalt:	Highway[city1=	Highway[city
	cost	X, city2=X,	1=2,
		cost=X.X,	city2=0,
		asphalt=X.X],	cost=7.0,
			asphalt=159.
		Highway[city1=	0],
		X, city2=X,	
		cost=X.X,	Highway[city
		asphalt=X.X]	1=1,
]	city2=2,
		List[cost=6.0,
			asphalt=72.0
		Highway[city1=],
		X, city2=X,	
		cost=X.X,	Highway[city
		asphalt=X.X],	1=3,
			city2=2,
		Highway[city1=	cost=12.0,
		X, city2=X,	asphalt=212.
		cost=X.X,	0],
		asphalt=X.X],	
			Highway[city
		Highway[city1=	1=3,
		X, city2=X,	city2=1,
		cost=X.X,	cost=10.0,

	aanhal+-V Vl	a amb a 1 + -1 1 2
	asphalt=X.X]	asphalt=112.
		0],
	AdjacencyList[
	City 0: ->	Highway[city
	Highway[city1=	1=0,
	X, city2= X ,	city2=3,
	cost=X.X,	cost=14.0,
	asphalt=X.X]	asphalt=415.
	->	0]
	Highway[city1=]
	X, city2= X ,	List[
	cost=X.X,	
	asphalt=X.X]	Highway[city
	->	1=1,
	Highway[city1=	city2=0,
	X, city2=X,	cost=5.0,
	cost=X.X,	asphalt=99.0
	asphalt=X.X]],
	City 1: ->	
	Highway[city1=	Highway[city
	X, city2=X,	1=1,
	cost=X.X,	city2=2,
	asphalt=X.X]	cost=6.0,
	->	asphalt=72.0
	Highway[city1=],
	X, city2=X,	
	cost=X.X,	Highway[city
	asphalt=X.X]	1=3,
	City 2: ->	city2=1,
	Highway[city1=	cost=10.0,
	X, city2=X,	asphalt=112.
	cost=X.X,	0]
	asphalt=X.X]]
	->	AdjacencyLis
	Highway[city1=	t[
	X, city2=X,	City 0:
	cost=X.X,	->
	asphalt=X.X]	Highway[city
	-1	5 . 5-7 [0 = 0]

->	1=0,
<pre>Highway[city1= X, city2=X,</pre>	city2=3, cost=14.0,
cost=X.X,	asphalt=415.
asphalt=X.X]	0] ->
City 3: ->	Highway[city
Highway[city1=	1=1,
X, city2=X,	city2=0,
<pre>cost=X.X, asphalt=X.X]</pre>	cost=5.0, asphalt=99.0
	->
-	Highway[city
	1=2,
	city2=0,
	cost=7.0,
	asphalt=159.
	0] City 1:
	->
	Highway[city
	1=1,
	city2=0,
	cost=5.0,
	asphalt=99.0
] -> Highway[city
	1=1,
	city2=2,
	cost=6.0,
	asphalt=72.0
] ->
	Highway[city
	1=3, city2=1,
	cost=10.0,
	asphalt=112.
	0]

City 2: -> Highway[city 1=1, city2=2, cost=6.0, asphalt=72.0] -> Highway[city 1=2,city2=0, cost=7.0, asphalt=159. 0] -> Highway[city 1=3,city2=2, cost=12.0, asphalt=212. 0] City 3: -> Highway[city 1 = 0, city2=3, cost=14.0, asphalt=415. 0] -> Highway[city 1 = 3, city2=1, cost=10.0, asphalt=112. 0] -> Highway[city 1=3, city2=2, cost=12.0,

ECP -	Run the program	ConsoleOut:	asphalt=212.
Invalid word checked	with no extra arguments. -Program asks for highway map: input/invalidmap.tx t	Invalid argument: negative number	Error: Invalid file! An IllegalArgum entException is thrown
BVA - Check word with multiple spell checking rules	Run the program without any extra arguments. -Program asks for highway map: input/invalidbounda rymap.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 IllegalArgum entException is thrown
DT - Illegal Characte rs in text file	Program is run without any extra arguments. -Program asks for highway map: input/illegalcharac ters.txt	Program stops after encountering error 'IllegalArgume nt', exits with the following output	Error: Invalid file! An IllegalArgum entException is thrown

	ConsoleOut:Inv	
	alid Argument:	
	Invalid	
	characters	