KEY: Orange Highlight = Test added during this milestone
Red Highlight = Impossible Test Case

Test	İ	ory Partition			
FilterGradesTest .testFilterByValu eLessThan	Categories: List length L Grade value V Number of existing grades fitting the filter G Test frames:				
	V:				
	V1)	< 0		[error]	
	V2)	0		[property nogrades]	
	V3)	1-100		[property somegrades]	
	V4)	101		[property allgrades]	
	V5)	> 101		[error]	
	L:				
	L1)	0		[if not error, property nogradesexist]	
	L2)	>0		[if not error, property gradesexist]	
	G:				
	G1)	No grades fit t	he	[if (gradesexist and (nogrades or somegrades)) or (nogradesexist and (somegrades or allgrades))]	
	G2)	Some grades the filter	fit	[if gradesexist and somegrades]	
	G3) All grades fit the filter		he	[if gradesexist and (somegrades or allgrades)]	
	Test cases:				
	Test Case ID		Variable values		
	V1		V =	-1	
	V2L1G1		V =	0, L = 0, G = 0	
	V2L2G1		V =	0, L = 5, G = 0	

V3L1G1	V = 50, L = 0, G = 0
V3L2G1	V = 20, L = 5, G = 0
V3L2G2	V = 85, L = 5, G = 3
V3L2G3	V = 98, L = 5, G = 5
V4L1G1	V = 101, L = 0, G = 0
V4L2G3	V = 101, L = 5, G = 5
V5	V = 102

FilterGradesTest .testFilterByValu eGreaterThan

- Categories:

 List length L

 Grade value V

 Number of existing grades fitting the filter G
 Test frames:

V:		
V1)	< -1	[error]
V2)	-1	[property allgrades]
V3)	0-99	[property somegrades]
V4)	100	[property nogrades]
V5)	> 100	[error]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No grades fit the filter	[if (gradesexist and (nogrades or somegrades)) or (nogradesexist and (somegrades or allgrades))]
G2)	Some grades fit the filter	[if gradesexist and somegrades]
G3)	All grades fit the filter	[if gradesexist and (somegrades or allgrades)]

Test Case ID	Variable values
V1	V = -2
V2L1G1	V = -1, L = 0, G = 0
V2L2G3	V = -1, L = 5, G = 5
V3L1G1	V = 89, L = 0, G = 0
V3L2G1	V = 98, L = 5, G = 0
V3L2G2	V = 89, L = 5, G = 2
V3L2G3	V = 20, L = 5, G = 5
V4L1G1	V = 100, L = 0, G = 0
V4L2G1	V = 100, L = 5, G = 0
V5	V = 101

FilterGradesTest .testFilterByValu eEqualTo

- Categories:

 List length L
 Grade value V
 Number of existing grades fitting the filter G

V:		
V1)	< 0	[error]
V2)	0-99	[property somegrades]
V3)	> 100	[error]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No grades fit the	[if nogradesexist or somegrades]

	filter	
G2)	Some grades fit the filter	[if gradesexist and somegrades]
G3)	All grades fit the filter	[if gradesexist and somegrades]

Test Case ID	Variable values
V1	V = -1
V2L1G1	V = 90, L = 0, G = 0
V2L2G1	V = 50, L = 5, G = 0
V2L2G2	V = 90, L = 5, G = 1
V2L2G3	V = 90, L = 5, G = 5
V3	V = 101

FilterGradesTest .testFilterByCat egory

- Categories:

 List length L
 Grade value V
 Number of existing grades fitting the filter G

V:		
V1)	Not existent in possible grades	[error]
V2)	Existent in possible grades	[validgrade]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No grades fit the	[if nogradesexist or validgrade]

	filter	
G2)	Some grades fit the filter	[if gradesexist and validgrade]
G3)	All grades fit the filter	[if gradesexist and validgrade]

Test Case ID	Variable values
V1	V = 'X'
V2L1G1	V = 'A', L = 0, G = 0
V2L2G1	V = 'F', L = 5, G = 0
V2L2G2	V = 'A', L = 5, G = 2
V2L2G3	V = 'A', L = 5, G = 5

FilterGradesTest .testFilterByOcc urencesScore

- Categories:

 List length L
 Number of occurrences N
 Number of existing grades fitting the filter G

< 1	[error]
> 1 && N <= L	[validnumber]
> L	[error]
	[if not error, property nogradesexist]
0	[if not error, property gradesexist]
o arades fit the	[if nogradesexist or validnumber]
	grades fit the

G2)	Some grades fit the filter	[if gradesexist and validnumber]
G3)	All grades fit the filter	[if gradesexist and validnumber]

Test Case ID	Variable values
N1	N = -1
N2L1G1	N = 1, L = 0, G = 0
N2L2G1	N = 2, L = 5, G = 0
N2L2G2	N = 1, L = 5, G = 5
N2L2G3	N = 5, L = 5, G = 5

FilterGradesTest .testFilterByOcc urencesGrade

Categories:

- List length LNumber of occurrences N
- Number of existing grades fitting the filter G

N:		
N1)	N < 1	[error]
N2)	N > 1 && N <= L	[validnumber]
N3)	N > L	[error]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No grades fit the filter	[if nogradesexist or validnumber]
G2)	Some grades fit the filter	[if gradesexist and validnumber]

	G3)	All grades fit the filter		[if gradesexist and validnumber]			
	Test cases:						
	Test 0	Case ID	Vari	iable values			
	N1		N =	-1			
	N2L1	G1	N =	1, L = 0, G = 0			
	N2L2	G1	N =	3, L = 5, G = 0			
	N2L2	G2	N =	2, L = 5, G = 2			
	N2L2	G3	N =	5, L = 5, G = 5			
	N3		N =	6			
FilterGradesTest .testShowResult s	Catego • Test fra	List length L					
	L:						
	L1)	0		[if not error, property nogradesexist]			
	L2)	>0		[if not error, property gradesexist]			
	Test ca	ases:					
	Test 0	Case ID	Vari	iable values			
	L1		L=	0			
	L2		L=	5			
FilterGradesTest .testNumOccure ncesDoubleList OfGrade	Categories: List length L Grade value to find occurrences of V Number of existing grades fitting the filter G Test frames:						

V:		
V1)	V < 0	[error]
V2)	0 <= V <= 100	[validvalue]
V3)	V > 100	[error]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No occurrences found	[if nogradesexist or validvalue]
G2)	Some occurrences found	[if gradesexist and validvalue]
G3)	Max occurrences found	[if gradesexist and validvalue]

Test Case ID	Variable values
V1	V = -1
V2L1G1	V = 90.1, L = 0, G = 0
V2L2G1	V = 85.0, L = 5, G = 0
V2L2G2	V = 90.1, L = 5, G = 1
V2L2G3	V = 90.1, L = 5, G = 5
V3	V = 101

FilterGradesTest .testNumOccure ncesCharacterLi stOfGrade

- Categories:

 List length L
 Grade to find occurrences of V
 Number of existing grades fitting the filter G

V:		
V1)	Not existent in possible grades	[error]
V2)	Existent in possible grades	[validvalue]
L:		
L1)	0	[if not error, property nogradesexist]
L2)	>0	[if not error, property gradesexist]
G:		
G1)	No occurrences found	[if nogradesexist or validvalue]
G2)	Some occurrences found	[if gradesexist and validvalue]
G3)	Max occurrences found	[if gradesexist and validvalue]

Test Case ID	Variable values
V1	V = -1
V2L1G1	V = 'A', L = 0, G = 0
V2L2G1	V = 'F', L = 5, G = 0
V2L2G2	V = 'A', L = 5, G = 2
V2L2G3	V = 'A', L = 5, G = 5

Test	Data Flow Coverage							
Filter	•							
Grade sTest.	Criterio	listToMa	filtervalu	filter	emptyLi	resultLis	Progra	Test

testFil terBy Value LessT han	n	nipulate	е		st	t	m Paths	Inputs
	All-def	<254, 256, 258, 260, {262, 264 266, 268}* 262>	<254, 256>	<256, 258, 260>	<258, 260>	<260, {262, 264, 266, 268}*, 262, 264, 266>	<254, 256, 258, 260, {262, 264, 266, 268}*, 262, 274>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 85)
	All-use	<254, 256, 258, 260, {262, 264 266, 268}* 262> <254, 256, 258, 260, {262, 264, 266, 268}* 262,	<254, 256>	<256, 258, 260>	<258, 260>	<260, {262, 264, 266, 268}*, 262, 264, 266> <260, {262, 264, 266, 268}*, 262, 274>	<254, 256, 258, 260, {262, 264, 266, 268}*, 262, 274>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 85)
	All-P-us er-some -C-uses	<254, 256, 258, 260, {262, 264 266, 268}* 262>	<254, 256>	<256, 258, 260>	<258, 260>	<260, {262, 264, 266, 268}*, 262, 264, 266>	<254, 256, 258, 260, {262, 264, 266, 268}*, 262, 274>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 85)
	All-C-us er-some -P-uses	<254, 256, 258, 260,	<254, 256>	<256, 258, 260>	<258, 260>	<260, {262, 264, 266,	<254, 256, 258, 260,	(listToM anipulat e = {grade1,

	{262, 264, 266, 268}* 262> <254, 256, 258, 260, {262, 264, 266, 268}* 262, 264, 266>				268}*, 262, 264, 266> <260, {262, 264, 266, 268}*, 262, 274>	{262, 264, 266, 268}*, 262, 274>	grade2, grade3, grade4, grade5}, filterVal ue = 85)
 All DU paths	<254, 256, 258, 260, {262, 264 266, 268}* 262> <254, 256, 258, 260, {262, 264 266, 268}* 262, 264, 266,	<254, 256>	<256, 258, 260>	<258, 260>	<260, {262, 264, 266, 268}*, 262, 264, 266, 264, 266, 268}*, 262, 274>	<254, 256, 258, 260, {262, 264, 266, 268}*, 262, 274>	(listToM anipulat e = {grade1, grade2, grade4, grade5}, filterVal ue = 85)

Filter Grade sTest. testFil terBy Value Great erTha n

•							
Criterio n	listToMa nipulate	filterval ue	filter	emptyLi st	resultLi st	Progra m Paths	Test Inputs
All-def	<277, 279, 281, 283, {285,	<277, 279>	<279, 281, 283>	<281, 283>	<283, {285, 287, 289, 291}*,	<277, 279, 281, 283, {285,	(listToM anipulat e = {grade1

	287, 289, 291}* 285>				285, 287, 289>	287, 289, 291}* 285, 297>	grade2, grade3, grade4, grade5} , filterVal ue = 89)
All-use	<277, 279, 281, 283, {285, 287, 289, 291}* 285> <277, 279, 281, 283, {285, 287, 289, 291}* 285,	<277, 279>	<279, 281, 283>	<281, 283>	<283, {285, 287, 289, 291}*, 285, 287, 289> <283, {285, 287, 289, 291}*, 297>	<277, 279, 281, 283, {285, 287, 289, 291}* 285, 297>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 89)
All-P-us er-some -C-uses	<277, 279, 281, 283, {285, 287, 289, 291}*	<277, 279>	<279, 281, 283>	<281, 283>	<283, {285, 287, 289, 291}*, 285, 287, 289>	<277, 279, 281, 283, {285, 287, 289, 291}* 285, 297>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 89)
All-C-us er-some -P-uses	<277, 279, 281, 283, {285,	<277, 279>	<279, 281, 283>	<281, 283>	<283, {285, 287, 289, 291}*,	<277, 279, 281, 283, {285,	(listToM anipulat e = {grade1

	287, 289, 291}* 285> <277, 279, 281, 283, {285, 287, 289, 291}* 285, 287, 289>				285, 287, 289> <283, {285, 287, 289, 291}*, 285, 297>	287, 289, 291}* 285, 297>	grade2, grade3, grade4, grade5} , filterVal ue = 89)
All D path:	,	<277, 279>	<279, 281, 283>	<281, 283>	<283, {285, 287, 289, 291}*, 285, 287, 289> <283, {285, 287, 289, 291}*, 295,	<277, 279, 281, 283, {285, 287, 289, 291}* 285, 297>	(listToM anipulat e = {grade1, grade2, grade4, grade5}, filterVal ue = 89)

Filter Grade sTest. testFil terBy Value Equal To

Criterio n	listToM anipulat	filterval ue	filter	emptyLi st	resultLi st	Progra m	Test Inputs
All-def	e <300, 302, 304, 306, {308,	<300, 302>	<302, 304, 306>	<304, 306>	<306, {308, 310, 312, 314}*,	<pre>>300, 302, 304, 306, {308,</pre>	(listToM anipulat e = {grade1

	310, 312, 314}* 308>				308, 310, 312>	310, 312, 314}* 308, 320>	grade2, grade3, grade4, grade5} , filterVal ue = 90)
All-use	<300, 302, 304, 306, {308, 310, 312, 314}* 308> <300, 302, 304, 306, {308, 310, 312, 314}* 308, 3112,	<300, 302>	<302, 304, 306>	<304, 306>	<306, {308, 310, 312, 314}*, 308, 310, 312> <306, {308, 310, 312, 314}*, 308, 320>	<300, 302, 304, 306, {308, 310, 312, 314}* 308, 320>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 90)
All-P-us er-som e-C-use s	<300, 302, 304, 306, {308, 310, 312, 314}* 308>	<300, 302>	<302, 304, 306>	<304, 306>	<306, {308, 310, 312, 314}*, 308, 310, 312>	<300, 302, 304, 306, {308, 310, 312, 314}* 308, 320>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 90)
All-C-us er-som e-P-use s	<300, 302, 304, 306, {308,	<300, 302>	<302, 304, 306>	<304, 306>	<306, {308, 310, 312, 314}*,	<300, 302, 304, 306, {308,	(listToM anipulat e = {grade1

	310, 312, 314}* 308> <300, 302, 304, 306, {308, 310, 312, 314}* 308, 310, 312>				308, 310, 312> <306, {308, 310, 312, 314}*, 308, 320>	310, 312, 314}* 308, 320>	grade2, grade3, grade4, grade5} , filterVal ue = 90)
All DU paths	<300, 302, 304, 306, {308, 310, 312, 314}* 308> <300, 302, 304, 306, {308, 310, 312, 314}* 308, 310, 312,	<300, 302>	<302, 304, 306>	<304, 306>	<306, {308, 310, 312, 314}*, 308, 310, 312> <306, {308, 310, 312, 314}*, 308, 320>	<300, 302, 304, 306, {308, 310, 312, 314}* 308, 320>	(listToM anipulat e = {grade1, grade2, grade3, grade4, grade5}, filterVal ue = 90)

Filter Grade sTest. testFil terBy Categ ory

•

Criterio n	listToM anipula te	filterCh aracter	filter	emptyL ist	resultLi st	Progra m Paths	Test Inputs
All-def	<323, 325, 327, 329, {331,	<323, 325>	<325, 327, 329>	<327, 329>	<329, {331, 333, 335, 337}*,	<323, 325, 327, 329, {331,	(listTo Manipu late = {grade 1,

	333, 335, 337}* 331>				331, 333, 335>	333, 335, 337}* 331, 343>	grade2 , grade3 , grade4 , grade5 }, filterCh aracter = 'A')
All-use	<323, 325, 327, 329, {331, 333, 335, 331> <323, 325, 327, 329, {331, 333, 335, 337}* 331, 333, 335,	<323, 325>	<325, 327, 329>	<327, 329>	<329, {331, 333, 335, 337}*, 331, 335> <329, {331, 333, 335, 337}*, 331, 343>	<323, 325, 327, 329, {331, 333, 335, 337}* 331, 343>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4 , grade5 }, filterCh aracter = 'A')
All-P-u ser-so me-C-u ses	<323, 325, 327, 329, {331, 335, 335, 337}* 331>	<323, 325>	<325, 327, 329>	<327, 329>	<329, {331, 333, 335, 337}*, 331, 333, 335>	<323, 325, 327, 329, {331, 335, 337,* 331, 343>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4 , grade5 }, filterCh aracter = 'A')

		All-C-u ser-so me-P-u ses	<323, 325, 327, 329, {331, 333, 335, 331> <323, 325, 327, 329, {331, 333, 335, 337}* 331, 335, 335,	<323, 325>	<325, 327, 329>	<327, 329>	<329, {331, 333, 335, 337}*, 331, 335> <329, {331, 333, 335, 337}*, 331, 343>	<323, 325, 327, 329, {331, 333, 335, 337}* 331, 343>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterCh aracter = 'A')
		All DU paths	<323, 325, 327, 329, {331, 333, 335, 337}* 331> <323, 325, 327, 329, {331, 333, 335, 337}* 331, 333, 335,	<323, 325>	<325, 327, 329>	<327, 329>	<329, {331, 333, 335, 337}*, 331, 335> <329, {331, 333, 335, 337}*, 331, 343>	<323, 325, 327, 329, {331, 335, 337}* 331, 343>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterCh aracter = 'A')
Filter	•								
Grade sTest. testFil terBy		Criterio n	listToM anipula te	filterA mount	filter	emptyL ist	resultLi st	Progra m Paths	Test Inputs
Occur									

ences Score	All-def	<346, 348, 350, 352, {354, 356, 358, 360}* 354>	<346, 348>	<348, 350, 352>	<350, 352>	<352, {354, 356, 358, 360}*, 354, 356, 358>	<346, 348, 350, 352, {354, 356, 358, 360}* 354, 366>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4 , grade5 }, filterA mount = 1)	
		All-use	<346, 348, 350, 352, {354, 356, 358, 360}* 354> <346, 348, 350, 352, {354, 356, 358, 360}* 358, 360}*	<346, 348>	<348, 350, 352>	<350, 352>	<352, {354, 356, 358, 360}*, 354, 356, 358> <352, {354, 356, 358, 360}*, 354, 366>	<346, 348, 350, 352, {354, 356, 358, 360}* 354, 366>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterA mount = 1)
		All-P-u ser-so me-C-u ses	<346, 348, 350, 352, {354, 356, 358, 360}* 354>	<346, 348>	<348, 350, 352>	<350, 352>	<352, {354, 356, 358, 360}*, 354, 356, 358>	<346, 348, 350, 352, {354, 356, 358, 360}* 354, 366>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4

							grade5
							filterA mount = 1)
All-C-u ser-so me-P-u ses	<346, 348, 350, 352, {354, 356, 358, 360}* 354> <346, 348, 350, 352, {354, 356, 358, 360}* 354, 356, 358,	<346, 348>	<348, 350, 352>	<350, 352>	<352, {354, 356, 358, 360}*, 354, 356, 358> <352, {354, 356, 358, 360}*, 354, 366>	<346, 348, 350, 352, {354, 356, 358, 360}* 354, 366>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterA mount = 1)
All DU paths	<346, 348, 350, 352, {354, 356, 358, 360}* 354> <346, 352, {354, 356, 358, 360}* 354, 356, 358,	<346, 348>	<348, 350, 352>	<350, 352>	<352, {354, 356, 358, 360}*, 354, 356, 358> <352, {354, 356, 358, 360}*, 366>	<346, 348, 350, 352, {354, 356, 358, 360}* 354, 366>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterA mount = 1)

Filter	•								
Grade sTest. testFil terBy	sTest. testFil terBy	Criterio n	listToM anipula te	filterA mount	filter	emptyL ist	resultLi st	Progra m Paths	Test Inputs
Occur ences Grade	All-def	<369, 371, 373, 375, {377, 379, 381, 383}* 377>	<369, 371>	<371, 373, 375>	<373, 375>	<375, {377, 379, 381, 383}*, 377, 379, 381>	<369, 371, 373, 375, {377, 379, 381, 383}* 377, 389>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterA mount = 2)	
		All-use	<369, 371, 373, 375, {377, 379, 381, 383}* 377> <369, 371, 373, 375, {377, 379, 381, 383}* 377, 381,	<369, 371>	<371, 373, 375>	<373, 375>	<375, {377, 379, 381, 383}*, 377, 381> <375, {377, 381, 383}*, 377, 389>	<369, 371, 373, 375, {377, 379, 381, 383}* 377, 389>	(listTo Manipu late = {grade 1, grade2, grade3, grade4, grade5 }, filterA mount = 2)
		All-P-u ser-so me-C-u ses	<369, 371, 373, 375, {377, 379,	<369, 371>	<371, 373, 375>	<373, 375>	<375, {377, 379, 381, 383}*, 377,	<369, 371, 373, 375, {377, 379,	(listTo Manipu late = {grade 1, grade2

Ι								
		381, 383}* 377>				379, 381>	381, 383}* 377, 389>	grade3 , grade4 , grade5 }, filterA mount = 2)
	All-C-u ser-so me-P-u ses	<369, 371, 373, 375, {377, 379, 381, 383}* 377> <369, 371, 373, 375, {377, 379, 381, 379, 381>	<369, 371>	<371, 373, 375>	<373, 375>	<375, {377, 379, 381, 383}*, 377, 379, 381> <375, {377, 379, 381, 383}*, 377, 389>	<369, 371, 373, 375, {377, 379, 381, 383}* 377, 389>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4 , grade5 }, filterA mount = 2)
	All DU paths	<369, 371, 373, 375, {377, 379, 381, 383}* 377> <369, 371, 373, 375, {377, 379, 381, 383}*	<369, 371>	<371, 373, 375>	<373, 375>	<375, {377, 379, 381, 383}*, 377, 379, 381> <375, {377, 379, 381, 383}*, 377, 389>	<369, 371, 373, 375, {377, 379, 381, 383}* 377, 389>	(listTo Manipu late = {grade 1, grade2 , grade3 , grade4 , grade5 }, filterA mount = 2)

										$\overline{1}$
			377, 379, 381>							
Filter Grade sTest. testSh	•	Criterion	resul	tList	resultStrin	gradeStri g	n Prog		Test Inputs	
owRe sults		All-def	396,	402, 406, ,	<394, 396, 398>	<396, 398 {400, 402 404, 406, 408}*, 400, 402	396, 400,	394, 398, 402>	(resultList = {grade1, grade2, grade3, grade4, grade5})	
		All-use	396, {400, 404, 408}* 400> <392 396,	402, 406, , 394, 398, 402, 406,	<394, 396, 398> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 402, 404, 406, 408}*, 400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411, 413>	<396, 398 {400, 402 404, 406, 408}*, 400, 402 <396, 398 {400, 406, 408}*, 400, 402, 404, 406, 408>	396, {400, 404, 408} 3, 400, 413>	, 402, 406, *, 411,	(resultList = {grade1, grade2, grade3, grade4, grade5})	
		All-P-use some-C-		, 394, 398,	<394, 396, 398>	<396, 398 {400, 402		394, 398,	(resultList = {grade1,	

	ses	{400, 402, 404, 406, 408}*, 400>		404, 406, 408}*, 400, 402>	400, 402>	grade2, grade3, grade4, grade5})
	All-C-user -some-P-u ses	<392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400> <392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400, 402>	<394, 396, 398> <394, 396, 398, {400, 402, 404, 406, 402, 404, 406, 408}*, 400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411> <394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411, 413>	<396, 398, {400, 402, 404, 406, 408}*, 400, 402, 404, 406, 408}*, 400, 402, 404, 406, 408>	<392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411, 413>	(resultList = {grade1, grade2, grade3, grade4, grade5})
	All DU paths	<392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400> <392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400, 402>	<394, 396, 398> <394, 396, 398, {400, 402, 404, 406, 402, 404> <394, 396, 398, {400, 402, 404, 406, 402, 404, 406, 408}*,	<396, 398, {400, 402, 404, 406, 408}*, 400, 402> <396, 398, {400, 402, 404, 406, 408}*, 400, 402, 404, 406, 408>	<392, 394, 396, 398, {400, 402, 404, 406, 408}*, 400, 411, 413>	(resultList = {grade1, grade2, grade3, grade4, grade5})

39 40 40 40 40 40 40 40 40 40 40 40 40 40	394, 396, 98, {400, 02, 404, 06, 08}*, 00, 411> 394, 396, 98, {400, 02, 404, 06, 08}*, 00, 411, 13>
--	---

Filter Grade sTest. testN umOc curen cesDo ubleLi stOfG rade

.

Criterion	scoreToC ount	allGrades	scoreCou nt	Program Paths	Test Inputs
All-def	<416, 418, {420, 422, 424}*, 420, 422>	<416, 418, {420, 422, 424}*, 420>	<418, {420, 422, 424}*, 420, 422, 424>	<416, 418, {420, 422, 424}*, 420, 422, 424>	(scoreToC ount = 90.1, allGrades = {grade1, grade2, grade3, grade4, grade5})
All-use	<416, 418, {420, 422, 424}*, 420, 422>	<416, 418, {420, 422, 424}*, 420> <416, 418, {420, 422, 424}*, 420, 422>	<418, {420, 422, 424}*, 420, 422, 424> <418, {420, 422, 424}*, 420, 428>	<416, 418, {420, 422, 424}*, 420, 428>	(scoreToC ount = 90.1, allGrades = {grade1, grade2, grade3, grade4, grade5})
All-P-user- some-C-u ses	<416, 418, {420, 422, 424}*, 420, 422>	<416, 418, {420, 422, 424}*, 420>	<418, {420, 422, 424}*, 420, 422, 424>	<416, 418, {420, 422, 424}*, 420, 422, 424>	(scoreToC ount = 90.1, allGrades = {grade1, grade2, grade3,

							grade4, grade5})
	All-C-user -some-P-u ses	<416, 418, {420, 422, 424}*, 420, 422>	<416, 418, {420, 422, 424}*, 420> <416, 418, {420, 422, 424}*, 420, 422>	<418, {420, 422, 424}*, 420, 422, 424> <418, {420, 422, 424}*, 420, 428>	<416, 418, {420, 422, 424}*, 420, 428>	(scoreToC ount = 90.1, allGrades = {grade1, grade2, grade3, grade4, grade5})	
	All DU paths	<416, 418, {420, 422, 424}*, 420, 422>	<416, 418, {420, 422, 424}*, 420> <416, 418, {420, 422, 424}*, 420, 422>	<418, {420, 422, 424}*, 420, 422, 424> <418, {420, 422, 424}*, 420, 428>	<416, 418, {420, 422, 424}*, 420, 428>	(scoreToC ount = 90.1, allGrades = {grade1, grade2, grade3, grade4, grade5})	
F:11							
Filter Grade sTest. testN	•	Criterion	scoreToC ount	allGrades	scoreCou nt	Program Paths	Test Inputs
umOc curen cesCh aracte rListO fGrad e	All-def	<431, 433, {435, 437, 439}*, 435, 437>	<431, 433, {435, 437, 439}*, 435>	<433, {435, 437, 439}*, 435, 437, 439>	<431, 433, {435, 437, 439}*, 435, 437, 439>	(scoreToC ount = 'A', allGrades = {grade1, grade2, grade3, grade4, grade5})	
		All-use	<431, 433, {435, 437, 439}*, 435, 437>	<431, 433, {435, 437, 439}*, 435> <431, 433, {435, 437, 439}*, 435, 437>	<433, {435, 437, 439}*, 435, 437, 439> <433, {435, 437, 439}*, 435, 428>	<431, 433, {435, 437, 439}*, 435, 428>	(scoreToC ount = 'A', allGrades = {grade1, grade2, grade3, grade4, grade5})
		All-P-user- some-C-u	<431, 433, {435, 437,	<431, 433, {435, 437,	<433, {435, 437,	<431, 433, {435, 437,	(scoreToC ount = 'A',

	ses	439}*, 435, 437>	439}*, 435>	439}*, 435, 437, 439>	439}*, 435, 437, 439>	allGrades = {grade1, grade2, grade3, grade4, grade5})
	All-C-user -some-P-u ses	<431, 433, {435, 437, 439}*, 435, 437>	<431, 433, {435, 437, 439}*, 435> <431, 433, {435, 437, 439}*, 435, 437>	<433, {435, 437, 439}*, 435, 437, 439> <433, {435, 437, 439}*, 435, 428>	<431, 433, {435, 437, 439}*, 435, 428>	(scoreToC ount = 'A', allGrades = {grade1, grade2, grade3, grade4, grade5})
	All DU paths	<431, 433, {435, 437, 439}*, 435, 437>	<431, 433, {435, 437, 439}*, 435> <431, 433, {435, 437, 439}*, 435, 437>	<433, {435, 437, 439}*, 435, 437, 439> <433, {435, 437, 439}*, 435, 428>	<431, 433, {435, 437, 439}*, 435, 428>	(scoreToC ount = 'A', allGrades = {grade1, grade2, grade3, grade4, grade5})