COMP105 – Assignment 1 Feedback

Username: sgtentic
Overall Mark: 100/100

Below you will find a test report for each function that you submitted. Each test is broken down into a number of test sets, that each try to test different aspects of your function. In parts A and B, marks are awarded for each test set based on the number of test cases that were passed. In part C, the marking is harsher: in order to get the marks for a test set the function must pass all of the tests within that set.

Question 1: char_to_int

Test Set	Argument	Output	Expected	Result	Mark
1	'1'	1	1	Correct	8/8
	'3'	3	3	Correct	
	'7'	7	7	Correct	
	'8'	8	8	Correct	
2	'0'	0	0	Correct	2/2

Mark: 10/10

Test Set 1: This test set checks basic functionality.

Test Set 2: Although zero cannot be a valid argument in part A or B, the specification in the handout does require this function to be able to handle zero.

Question 2: repeat_char

Test Set	Argument	Output	Expected	Result	Mark
1	'a' 0	""	""	Correct	4/4
	'b' 1	"b"	"b"	Correct	
2	'c' 3	"ccc"	"ccc"	Correct	6/6
	'd' 7	"ddddddd"	"ddddddd"	Correct	
	'e' 13	"eeeeeeeeeeeee"	"eeeeeeeeeeeee"	Correct	
	'f' 123	"fffffffffffffff...	"fffffffffffffff...	Correct	

Mark: 10/10

Test Set 1: This test set checks whether the base case is correct. Note that zero is a valid input to this function, and the expected output would be zero copies of c, so the empty string.

Test Set 2: This test set checks whether the function outputs the correct number of characters.

Question 3: decode

Test Set	Argument	Output	Expected	Result	Mark
1	""	""	""	Correct	4/4
	"a1"	"a"	"a"	Correct	
2	"a3b3c3"	"aaabbbccc"	"aaabbbccc"	Correct	6/6
	"h1e1l2o1"	"hello"	"hello"	Correct	
	"d9e9f9g9h9"	"ddddddddeeeeeee...	"ddddddddeeeeeee...	Correct	
	"a1a2a3"	"aaaaaa"	"aaaaaa"	Correct	

Mark: 10/10

Test Set 1: This test set checks whether the base case is correct. Note that an empty string is a valid input to this function, and the expected output would be an empty string.

Test Set 2: This test set checks whether the function correctly decodes strings.

Question 4: int_to_char

Test Set	Argument	Output	Expected	Result	Mark
1	2	'2'	'2'	Correct	4/4
	4	'4'	'4'	Correct	
	6	'6'	'6'	Correct	
	9	'9'	'9'	Correct	
2	0	'0'	'0'	Correct	1/1

Mark: 5/5

Test Set 1: This test set checks basic functionality.

Test Set 2: Note that, although encode will never call this function with zero, the handout specifies that the function must work for zero as well.

Question 5: length_char

Test Set	Argument	Output	Expected	Result	Mark
1	'a' ""	0	0	Correct	2/2
	'a' "bbb"	0	0	Correct	
2	'a' "a"	1	1	Correct	4/4
	'c' "caaa"	1	1	Correct	
	'd' "dddeee"	3	3	Correct	
	'e' "eee"	3	3	Correct	
3	'f' "fgfgfgfg"	1	1	Correct	4/4
	'a' "aaabbbaaa"	3	3	Correct	

Mark: 10/10

Test Set 1: This test set checks the base cases of the function. Note that if the character c does not appear at the start of the list, then the answer should be zero.

Test Set 2: This test set checks basic functionality.

Test Set 3: This test set checks that the function only counts characters at the start of the list, and not every character in the list.

Question 6: drop_char

Test Set	Argument	Output	Expected	Result	Mark
1	'a' ""	""	""	Correct	2/2
	'a' "bbb"	"bbb"	"bbb"	Correct	
2	'a' "a"	""	""	Correct	4/4
	'c' "caaa"	"aaa"	"aaa"	Correct	
	'd' "dddeee"	"eee"	"eee"	Correct	
	'e' "eee"	""	""	Correct	
3	'f' "fgfgfgfg"	"gfgfgfg"	"gfgfgfg"	Correct	4/4
	'a' "aaabbbaaa"	"bbbaaa"	"bbbaaa"	Correct	

Mark: 10/10

Test Set 1: This test set checks the base cases of the function. Note that if the character c does not appear at the start of the list, then the function should not drop any characters.

Test Set 2: This test set checks basic functionality.

Test Set 3: This test set checks that the function only drops characters at the start of the list, and not every character in the list.

Question 7: encode

Test Set	Argument	Output	Expected	Result	Mark
1	`""`	`""`	`""`	`Correct`	4/4
	`"a"`	`"a1"`	`"a1"`	`Correct`	
2	`"aaabbbaaa"`	`"a3b3a3"`	`"a3b3a3"`	`Correct`	6/6
	`"hello"`	`"h1e1l2o1"`	`"h1e1l2o1"`	`Correct`	
	`"abcdefgh"`	`"a1b1c1d1e1f1g1h1"`	`"a1b1c1d1e1f1g1h1"`	`Correct`	
	`"ddddddddd"`	`"d9"`	`"d9"`	`Correct`	

Mark: 10/10

Test Set 1: This test set checks the base cases of the function. Note that an empty string is a valid input to the function, and its encoding is an empty string.

Test Set 2: This test set checks that the function can correctly encode strings.

Question 8: complex_encode

Test Set	Argument	Output	Expected	Result	Mark
1	`""`	`""`	`""`	`Correct`	1/1
2	`"aaaaaaaaaaaaaaa"`	`"a15"`	`"a15"`	`Correct`	2/2
	`"bbbbbbbbbbbbbbb..."`	`"b42"`	`"b42"`	`Correct`	
	`"ccccccccccccccc..."`	`"c99"`	`"c99"`	`Correct`	
3	`"ddddddddddddddd..."`	`"d123"`	`"d123"`	`Correct`	2/2
	`"eeeeeeeeeeeeeee..."`	`"e4567"`	`"e4567"`	`Correct`	
	`"fffffffffffffff..."`	`"f99999"`	`"f99999"`	`Correct`	
	`"ggggggggggggggg..."`	`"g1234567"`	`"g1234567"`	`Correct`	
4	`"a"`	`"a"`	`"a"`	`Correct`	2/2
	`"abcdef"`	`"abcdef"`	`"abcdef"`	`Correct`	
	`"aaabcccdeeef"`	`"a3bc3de3f"`	`"a3bc3de3f"`	`Correct`	
5	`"aaaaaaaaaaaaaaa..."`	`"a99bc99d"`	`"a99bc99d"`	`Correct`	3/3
	`"baaaaaaaaaaaaaa..."`	`"ba123b"`	`"ba123b"`	`Correct`	
	`"aaaaaaaaaaaaaaa..."`	`"a123456b"`	`"a123456b"`	`Correct`	

All tests passed bonus: 5/5

Mark: 15/15

Test Set 1: This test set checks the base cases of the function. Note that an empty string is a valid input to the function, and its encoding is an empty string.

Test Set 2: This test set checks whether the function can handle more than 9 repetitions.

Test Set 3: This test set checks whether the function can handle any number of repetitions. Note that the handout gave no limit on the number of repetitions, so the function should work no matter how many repetitions it sees.

Test Set 4: This test set checks whether your function correctly encodes lone-characters in strings that contain no more than nine repetitions.

Test Set 5: This test set checks whether your function correctly encodes lone-characters in any string

Question 9: complex_decode

Test Set	Argument	Output	Expected	Result	Mark
1	""	""	""	Correct	1/1
2	"a16"	"aaaaaaaaaaaaaaaa"	"aaaaaaaaaaaaaaaa"	Correct	2/2
	"b32"	"bbbbbbbbbbbbbbbb...	"bbbbbbbbbbbbbbbb...	Correct	
	"c64"	"cccccccccccccccc...	"cccccccccccccccc...	Correct	
3	"d256"	"ddddddddddddddddd...	"ddddddddddddddddd...	Correct	2/2
	"e1024"	"eeeeeeeeeeeeeeee...	"eeeeeeeeeeeeeeee...	Correct	
	"f65536"	"ffffffffffffffff...	"ffffffffffffffff...	Correct	
	"g131072"	"gggggggggggggggg...	"gggggggggggggggg...	Correct	
4	"a"	"a"	"a"	Correct	2/2
	"abcdef"	"abcdef"	"abcdef"	Correct	
	"a9bcda9e"	"aaaaaaaaabcdaaaa...	"aaaaaaaaabcdaaaa...	Correct	
5	"a9ba99ca9d"	"aaaaaaaaabaaaaaa...	"aaaaaaaaabaaaaaa...	Correct	3/3
	"ba999ba99ba9ba"	"baaaaaaaaaaaaaaa...	"baaaaaaaaaaaaaaa...	Correct	
	"a111111b"	"aaaaaaaaaaaaaaaa...	"aaaaaaaaaaaaaaaa...	Correct	

All tests passed bonus: 5/5
Mark: 15/15

Test Set 1: This test set checks the base cases of the function. Note that an empty string is a valid input to the function, and it decodes to an empty string.

Test Set 2: This test set checks whether the function can handle more than 9 repetitions.

Test Set 3: This test set checks whether the function can handle any number of repetitions. Note that the handout gave no limit on the number of repetitions, so your function should work no matter how many repetitions it sees.

Test Set 4: This test set checks whether the function correctly decodes lone-characters in strings that contain no more than nine repetitions.

Test Set 5: This test set checks whether the function correctly decodes lone-characters in any string

Perfect Part C bonus: 5/5

Overall Mark 100/100