Collated list of EXTREME test cases for 42 Minishell Project

[mts] - multiple tabs and spaces (To insert a tab character in Bash, press ctrl+v followed by Tab)

****All output will be assumed to have the newline character '\n' at the end unless explicitly stated otherwise****

Description	Command	Bash Output	Exit value	Notes
Unset the PATH variable before executing the echo tests. The echo command MUST be implemented as a built-in feature and not via execve with the echo binary. unset PATH				
Echo command with multiple "-n" options with multiple n, and an option at the end with characters other than 'n'	[mts]echo[mts]-n[mts]-nn[mts]-nnn[mts]-nnnn[mts]-nanny	-nanny (no newline)	0	
Echo command with multiple "-n"	[mts]echo[mts]-n[mts]-nn[mts]-nanny[mts]-nnnn	-nanny -nnnn -nnn (no newline)	0	

options with multiple n, and an option in the middle with characters other than ‘n’	[mts]-nnn			
Echo command with multiple “-n” options with multiple n, and an option at the start with characters other than ‘n’	[mts]echo[mts]-nanny[mts]-n[mts]-nn[mts]-nnn	-nanny -n -nn -nnn	0	
Echo command with parameter “--n”	[mts]echo[mts]--n	–n	0	
Echo command with parameter “-”	[mts]echo[mts]-	-	0	
Echo command with only	[mts]echo[mts]		0	

trailing spaces and tabs				
Echo command with “echo” as a parameter	[mts]echo[mts]echo	echo	0	
Unset the PATH and PWD environment variables before commencing the cd tests. unset PATH unset PWD After each test, check that $PWD is updated to reflect the current directory and $OLDPWD is blank. Make sure to unset PWD again before moving to the next test. Check notes for exceptions.				
Cd command with parameter “..”	[mts]cd[mts]..	(changes current directory to parent directory if not at root directory, otherwise does nothing)	0	
Cd command with parameter “.”	[mts]cd[mts].	(Nothing happens)		
Cd	[mts]cd[mts]	(changes	0	

command with no parameter		current directory to $HOME)		
Cd command with no parameter and no HOME environmen t variable	unset HOME [mts]cd[mts]	cd: HOME not set	1	$PWD will still remain blank
Cd command with more than 1 parameter	[mts]cd[mts] .[mts].	cd: too many arguments	1	
Cd command with nonexistent directory	[mts]cd[mts] abc	cd: abc: No such file or directory	1	
Cd command with something that is not a directory	[mts]cd[mts] /usr/bin/ls	Cd: /usr/bin/ls: Not a directory	1	
Unset the PATH and PWD environment variables before commencing the pwd tests. The pwd command MUST be implemented as a built-in and must still work even if $PATH and $PWD is blank. unset PATH				

unset PWD				
pwd command with no parameters	[mts]pwd[mts]	(The current directory must be displayed)	0	
Pwd command with more than 1 parameter	[mts]pwd[mts]pwd	(The current directory must be displayed)	0	
Export tests				
Export command with no parameters	[mts]export[mts]	(a long list of declare statements)	0	Your minishell does not have to show the same output as bash. How you want to handle this is subjective. An error message is acceptable.
Export command with 1 parameter, parameter variable doesn't	[mts]export[mts]bigass	Command 'env | grep bigass' should show nothing	0	

exist, new value not defined				
Export command with 1 parameter, parameter variable doesn't exist, '=' sign present, new value not defined	[mts]export[mts]bigass=	Command 'env | grep bigass' should show: bigass=	0	
Export command with 1 parameter, parameter variable doesn't exist, '=' sign present, new value is '='	[mts]export[mts]bigass==	Command 'env | grep bigass' should show: bigass==	0	
Export command with 1 parameter, parameter variable doesn't	[mts]export[mts]bigass[mts]=1	Export: `=1': not a valid identifier	1	Make sure that 'echo $bigass' displays nothing

exist, ‘=’ sign present but is separated by mts, new value defined.				
Export command with 1 parameter, parameter variable doesn’t exist, ‘=’ sign present, new value is defined but separated by mts.	[mts]export[mts]bigass=[mts]1	Command ‘env | grep bigass’ should show: bigass=	0	
Normal export command with 1 parameter, but value is 10 million characters long	[mts]export[mts]bigass=[10 million chars]	Command ‘echo $bigass’ should display 10 million chars	0	Not sure if minishell should handle this
Normal export command	[mts]export[mts]a=1[mts]b=2[mts]c=	(variables a, b and c added to	0	

with 3 parameters	3	env with values 1, 2 and 3 respectively)		
Export command with 3 parameters, but the 2nd and 3rd one overwrite the 1st one	[mts]export[mts]a=1[mts]a=2[mts]a=3	Command ‘env | grep a’ should show: a = 3	0	
Export command with 3 parameters, 2nd one has invalid syntax, 3rd one overwrites 1st one	[mts]export[mts]a=1[mts]b[mts]=2[mts]a=3	Export: `=2’: not a valid identifier Command ‘env | grep a’ should show: a = 3	1	
Export command with 3 parameters, 1st and 2nd have invalid syntax	[mts]export[mts]a[mts]=1[mts]b[mts]=2[mts]a=3	Export: `=1’: not a valid identifier Export: `=2’: not a valid identifier Command ‘env | grep a’ should show:	1	

		a = 3		
Unset Tests				
Unset command with no arguments	[mts]unset[mts]		0	
Unset command with multiple arguments	[mts]unset[mts]USER[mts]PWD	Command ‘env | grep USER’ should show nothing. Command ‘env | grep PWD’ should show nothing	0	
Unset command with 10 million arguments	[mts]unset[10 million arguments]		0	Not sure if minishell should handle this
Unset command with SHLVL as argument, at $SHLVL=6	First, run ./minishell until $SHLVL is 6. [mts]unset[mts]SHLVL	Command ‘env | grep SHLVL’ should show: SHLVL=0 Command ‘echo $SHLVL’	0	Using the exit command should bring minishell down to SHLVL=5

		should just display a newline		
Exit Tests				
Exit command with 1 numeric parameter, parameter has ‘+’ sign at start	[mts]exit[mts]+5	exit	5	
Exit command with 1 non-numeric parameter	[mts]exit[mts]++5	exit exit: ++5: numeric argument required	2	Bash will still exit
Exit command with 2 numeric arguments	[mts]exit[mts]1[mts]1	exit exit: too many arguments	1	Bash will not exit
Exit command with 2 arguments, 1st one is a non-numeric argument, 2nd one is a numeric argument	[mts]exit[mts]++5[mts]1	exit exit: ++5: numeric argument required	2	Bash will still exit

Exit command with LONG_MAX as an argument	[mts]exit[mts]9223372036854775807	exit	255	
Exit command with LONG_MIN as an argument	[mts]exit[mts]-9223372036854775808	exit	0	
Exit command with LONG_MAX+1 as an argument	[mts]exit[mts]9223372036854775808	exit exit: 9223372036854775808: numeric argument required	2	Bash will still exit
Exit command with LONG_MIN -1 as an argument	[mts]exit[mts]-9223372036854775809	exit: Exit: -9223372036854775809: numeric argument required	2	Bash will still exit