qShell
Test Cases

Shell Invocation

No.	Description	Input	Expected Output	Actual Output	Result
1.	Invoke qShell from Ash shell	# qsh	qShell should get invoked with a welcome message and shell prompt should appear	# qsh Welcome to qShell! ~$	**Pass**

Profile file

Default values for PROMPT, PATH, HOME and ALARM are -
PROMPT=$
HOME=/home/
PATH=/bin;/usr/bin
ALARM=ENABLED

Contents of the PROFILE file -
# cat /home/profile
PROMPT=$$
HOME=/home/dhanashree
PATH=/bin:/usr/bin
ALARM=DISABLED

No.	Description	Input	Expected Output	Actual Output	Result
1.	Profile file for shell is present when shell is being invoked.	# qsh	Values for PROMPT, HOME, PATH and ALARM should be set as specified in the profile file.	# qsh Welcome to qShell! ~$$ ~$$ pwd /home/dhanashree	**Pass**
2.	If profile file is not present then defaults	rm /home/profile	Defaults should be	# qsh Welcome to qShell!	**Pass**

	for PROMPT, PATH, HOME and ALARM should be set.	qsh	used.	~$ ~$ pwd /home	

3.	Invalid option (anything except PATH or PROMPT or HOME or ALARM) is specified in profile file.	# cat /home/profile
		PROMPT=$$
		HOME=/home/dhanashree
		ALARM=DISABLED
		INVALIDOPTION=invalid
			Values specified for all valid options should be set and invalid option is displayed on the screen.	# qsh
				
				Welcome to qShell!
				
				profile: unrecognised symbol INVALIDOPTION
				
				~$$ pwd
				/home/dhanashree
					Pass

4.	If some of the options are missing from the profile file then use default values for them.	# cat /home/profile
		PROMPT=$$
		PATH=/bin:/usr/bin
		ALARM=DISABLED
			Default value for HOME should be used.	# qsh
				
				Welcome to qShell!
				
				~$$ pwd
				/home
					Pass

5.	If value for one of the option is missing.	# cat /home/profile PROMPT= PATH=/bin:/usr/bin ALARM=DISABLED	A notice should displayed for missing value and default value for PROMPT should be used.	# qsh Welcome to qShell! profile: value not present for PROMPT; using default value ~$	**Pass**

Command Execution (with and without arguments)

No.	Description	Input	Expected Output	Actual Output	Result
1.	Invoke any command present in /bin or /usr/bin	ls	Contents of the current directory should be displayed.	~$ ls .profile.swp ast bin dhanashree qShell	**Pass**
2.	Invoke any command (with valid arguments) present in /bin or /usr/bin.	uname -r	Should display the Minix kernel version.	~$ uname -r 3.2.1	**Pass**

3.	Invoke any command (with invalid argument) present in /bin or /usr/bin.	ls -y	ls should error out with "invalid option" message.	~$ ls -y ls: unknown option -- y usage: ls [-AaBbCcdFfghikLlMmnopqRrSsTtuWwx1] [file ...]	**Pass**
4.	Invoke an invalid command does not exist in specified PATH	invalidcommand	Shell should error out as "command not found"	~$ invalidcommand invalidcommand Command not found in PATH.	**Pass**

Change Directory and Command Prompt

mkdir /home/dhanashree
mkdir /home/dhanashree/test

No.	Description	Input	Expected Output	Actual Output	Result
1.	Invoke shell and check the prompt sign.	# qsh	"~" should be displayed in the command prompt as a symbol for home directory.	# qsh Welcome to qShell! ~$	**Pass**
2.	Change to a directory which is specified by the absolute path.	cd /home/dhanashree/test	The directory should be changed to /home/dhanashree/test and prompt should be updated to display the basename of the (now)current dir.	~$ cd /home/dhanashree/test test$ test$ pwd /home/dhanashree/test	**Pass**
3.	Change to home directory using "~".	cd ~	The current directory should be changed to the one specified by profile file (if present) else to default home directory	test$ cd ~ ~$ pwd /home	**Pass**

			and prompt should display.		
4.	Change to a directory which is specified by absolute path starting with "~/".	cd ~/dhanashree	Current directory should be changed to /home/dhanashree and the basename of current directory should be displayed on the prompt.	test$ cd ~/dhanashree dhanashree$	**Pass**
5.	Change to a directory specified by relative path starting with the directory name directly.	cd test	Since this is relative path ,the current directory should be changed to a directory which is formed by pwd + relative path; and prompt should be updated to display the basename of the (now)current directory.	dhanashree$ cd test test$	**Pass**
6.	Change to a directory specified by relative path starting with "./" and then directory name.	cd ./test	Since this is relative path ,the current directory should be changed to a dir which is formed by pwd + relative path; and prompt should be updated to display the basename of the (now)current directory.	dhanashree$ cd ./test test$	**Pass**

7.	Going one level up the directory hierarchy	cd ..	The directory should be changed to the immediate parent directory of the current directory and the prompt should be updated to display the basename of the (now)current directory.	test$ cd .. dhanashree$	**Pass**
8.	Going multiple levels up the directory hierarchy.	cd ../..	The directory should be changed to the indirect parent directory specified by the number of ".."s and the prompt should be updated to display the basename of the (now)current directory.	test$ cd ../.. ~$	**Pass**
9.	Change to a path leading to the invalid directory.	cd /home/dhanashree/abcdssds	Shell should error out as "no such file or directory" and the current working directory should remain unchanged so should the prompt display.	~$ cd /home/dhanashree/abcdssds cd : no such file or directory ~$	**Pass**
10.	Change to a directory specified by absolute path having ".." in them.	cd /home/dhanashree/test/../test/	The directory should be changed to the one specified by the absolute path and the prompt should be updated to display the	~$ cd /home/dhanashree/test/../test/ test$	**Pass**

			basename of the (now)current directory.		
11.	No arguments to cd	cd	The current directory should be changed to the one specified by profile file (if present) else to default home dir and prompt should display ~	test$ cd ~$ pwd /home ~$	**Pass**

Alias for command

No.	Description	Input	Expected Output	Actual Output	Result
1.	Create alias for a command on the shell using "alias" command.	alias list=ls	An alias, list, should be created for command ls and when list is issued on the shell, the ls should get executed.	~$ alias list=ls ~$ list .profile.swp ALIAS_PROFILE ast bin dhanashree profile qShell sample tmp.out	**Pass**
2.	Restart the shell and use the alias created earlier in #1 above.	list	Shell should remember the alias created before and resolve them appropriately to actual command when issued.	~$ ^CDo you want to exit from qShell? (y/n) y # qsh ~$ list ALIAS_PROFILE ast bin dhanashree profile	**Pass**
3.	Use the alias with argument.	list -l	Appropriate arguments should get passed to the actual command. Here "ls -l" should	~$ list -l total 96 -rw-r--r-- 1 root operator 12288 Sep 27 20:36 .profile.swp -rw-r--r-- 1 root	**Pass**

			get executed.	operator 8 Sep 28 00:21 ALIAS_PROFILE drwxr-xr-x 2 ast operator 320 Feb 15 2013 ast drwxr-xr-x 2 bin operator 320 Feb 15 2013 bin drwxr-xr-x 3 root operator 192 Sep 27 23:31 dhanashree -rwxr-xr-x 1 root operator 40 Sep 27 23:27 profile	
4.	Use the I/O redirection with alias.	list -l > /tmp/list.out	The list should get resolved to ls and output of "ls -l" should get redirected to /tmp/list.out	~$ list -l > list.out ~$ cat /tmp/list.out total 96 -rw-r--r-- 1 root operator 12288 Sep 27 20:36 .profile.swp -rw-r--r-- 1 root operator 8 Sep 28 00:21 ALIAS_PROFILE drwxr-xr-x 2 ast operator 320 Feb 15 2013 ast drwxr-xr-x 2 bin operator 320 Feb 15 2013 bin drwxr-xr-x 3 root operator 192 Sep 27 23:31 dhanashree -rwxr-xr-x 1 root operator 40 Sep 27 23:27 profile	**Pass**
5.	Alias name is the name of the existing system command.	alias ps="ls"	An error should be thrown as there is already a system command existing with the same name as that of the alias.	~$ alias ps="ls" Error: Alias is a system command Cannot add to alias ~$	**Pass**
6.	Create alias for an invalid command.	alias foo="invalidcommand"	An error should be thrown specifying that the	~$ alias foo="invalidcommand" Error: Command not present	**Pass**

			command for which alias is being created does not exist.	Cannot add to alias ~$	

Conditional Command Execution Using ‘&&’ and ‘||’

No.	Description	Input	Expected Output	Actual Output	Result
1.	Combine commands using &&.	uname -p && ls	uname -p will be executed and if it completed with success then ls should be executed.	~$ uname -p && ls i386 ALIAS_PROFILE ast bin dateop.out dhanashree profile ~$	**Pass**
2.	Combine commands using ||.	uname -p || ls	uname -p will be executed and if it is not completed with success then ls should be executed else ls should not get executed.	~$ uname -p || ls i386 ~$	**Pass**
3.	Combine commands using both && and ||.	ls && uname -u || pwd	ls should first get executed and if it is successful then uname -u should get executed; As uname -u will fail since -u is an invalid option, the pwd should be executed.	~$ ls && uname -u || pwd ALIAS_PROFILE ast bin dateop.out dhanashree profile Usage: uname -snrvmpa /home ~$	**Pass**

Handling Ctrl + c

Alarm is disabled in the profile file
# cat /home/profile
PROMPT=$
HOME=/home/dhanashree
PATH=/bin:/usr/bin
ALARM=DISABLED

No.	Description	Input	Expected Output	Actual Output	Result
1.	Issue Ctrl + C for a command which is running or waiting.	wc -l Issue Ctrl + C	wc -l will wait for some input to work on. When Ctrl + C will be issued, the command should get killed and the prompt should reappear.	~$ wc -l ^C ~$	**Pass**
2.	Issue Ctrl + C when no command is being run.	Issue Ctrl + C on the prompt.	As no command is being run, on issuing Ctrl + C the shell should ask if user wants to exit from the shell(options should be y or n); if yes then shell should get terminated and Minix Ash shell prompt should appear else the qShell prompt should appear and shell should stay alive.	~$ ^CDo you want to exit from qShell? (y/n) y #	**Pass**
3.	Issue Ctrl + C when no command is being run and provide invalid option.	Issue Ctrl + C and provide 'g' when asked	As no command is being run, on issuing Ctrl + C the shell should ask if user wants to exit from the shell(options should be y or n); as 'g' is	~$ ^CDo you want to exit from qShell? (y/n) g Unrecognised option g	**Pass**

			invalid option, shell should error out and should not terminate.		
4.	Terminate shell by typing "exit" on the prompt.	exit	A confirmation message will be displayed on the screen and the shell should get terminated if user provides 'y' or 'Y' option and Minix Ash shell prompt should appear.	~$ exit Do you want to exit from qShell? (y/n) y #	**Pass**

I/O Redirection

No.	Description	Input	Expected Output	Actual Output	Result
1.	Redirect the output of a command to a file which does not exist.	date > dateop.out	The output of date command should get redirected to dateop.out file and nothing should be displayed on the screen. A new file dateop.out should get created in current directory.	~$ date > dateop.out ~$ ls ALIAS_PROFILE ast bin dateop.out dhanashree profile ~$ cat dateop.out Sun Sep 28 10:12:14 GMT 2014	**Pass**
2.	Redirect the output of a command to a file which already exists.	who > dateop.out	The output of who command should get redirected to the existing dateop.out file and nothing should be displayed on the screen. The contents of the dateop.out should get	~$ who > /home/dateop.out ~$ cat /home/dateop.out root console Sun Sep 28 09:32 root ttyp0 Sun Sep 28 09:32 (192.168.7.1)	**Pass**

			overwritten.		
3.	Append the output of a command to already existing file.	df >> /home/dateop.out	The output of df command should get redirected to the existing dateop.out file and nothing should be displayed on the screen. The existing contents of the dateop.out should remain and new contents should be appended.	~$ df >> /home/dateop.out ~$ cat /home/dateop.out root console Sun Sep 28 09:32 root ttyp0 Sun Sep 28 09:32 (192.168.7.1) Filesystem Size (kB) Free Used % Files% Mounted on /dev/c0d0p0s0 65536 45716 19820 31% 3% / none 0 0 0 0% 0% /proc /dev/c0d0p0s2 8446652 6245008 2201644 27% 3% /usr /dev/c0d0p0s1 1970176 1939180 30996 2% 1% /home none 0 0 0 0% 0% /sys	**Pass**
4.	Redirect the input for a command.	wc -l < /tmp/wcinput.out	wc -l should not wait for input to be typed on the terminal(stdin) instead it should consume the input from /tmp/wcinput.out and provide result for that.	~$ cat /tmp/wcinput.out Hi There !! This is our new shell !! ~$ wc -l < /tmp/wcinput.out 2	**Pass**
5.	Redirect the input for a command using an invalid file (which does not exist).	wc -l < inv.out	An error should be thrown since the file from which the inout is to be read, does not exist. The	~$ wc -l < inv.out no such file or directory ~$	**Pass**

			command should not execute and shell prompt should reappear.		
6.	Multiple redirections in a single command (input first then output).	wc -l < /tmp/wcinput.out > /tmp/wcoutput.out	wc -l should not wait for input to be typed on the terminal(stdin) instead it should consume the input from /tmp/wcinput.out and the result produced should get redirected to /tmp/wcoutput.out.	~$ cat /tmp/wcinput.out Hi There !! This is our new shell !! ~$ wc -l < /tmp/wcinput.out > /tmp/wcoutput.out ~$ cat /tmp/wcoutput.out 2 ~$	**Pass**
7.	Multiple redirections in a single command (output first then input).	wc -l > /tmp/wcoutput.out < /tmp/wcinput.out	wc -l should not wait for input to be typed on the terminal(stdin) instead it should consume the input from /tmp/wcinput.out and the result produced should get redirected to /tmp/wcoutput.out.	~$ rm /tmp/wcoutput.out ~$ wc -l > /tmp/wcoutput.out < /tmp/wcinput.out ~$ cat /tmp/wcoutput.out 2 ~$	**Pass**

Executing Process In The Background

No.	Description	Input	Expected Output	Actual Output	Result
1.	Put a command in the background.	uname -r &	The command should be run in the background and shell prompt should reappear (it	~$ uname -r & [1130] ~$ 3.2.1 ~$	**Pass**

			should not wait for the command to complete).		
2.	Put combined commands (combined with &&) in the background.	ls && pwd &	The commands should be executed in the background and both ls and pwd should get executed. Shell prompt should reappear as soon as the commands are entered(it should not wait for the command to complete).	~$ ls && pwd & [1053] [1054] ~$ ALIAS_PROFILE ast bin dateop.out dhanashree profile /home ~$	**Pass**
3.	Put combined commands (combined with ||) in the background.	ls || pwd &	The commands should be executed in the background and only ls should get executed. Shell prompt should reappear as soon as the commands are entered(it should not wait for the command to complete).	~$ ls || pwd & [1055] ~$ ALIAS_PROFILE ast bin dateop.out dhanashree profile ~$	**Pass**

Alarm

By default ALARM feature will be enabled. It can be ENABLED/DISABLED by specifying it accordingly in the profile file. The alarms can also be enabled or disabled from the command line using "alarmon" and "alarmoff" commands.

No.	Description	Input	Expected Output	Actual Output	Result
1.	Alarm is enabled and command exceeds the timeout.	sleep 8	A message should appear after 5 seconds asking	~$ sleep 8 Do you want to stop the program? (y/n) y Killing process	**Pass**

			whether you want to kill the command. If user enters 'y' or 'Y' then current process should get killed and prompt should reappear. And if user enters 'n' or 'N' then shell should let the current command complete and then the prompt should appear.	~$ ~$ sleep 6 Do you want to stop the program? (y/n) n Not killing process. Waiting for it to complete ~$	
2.	Enable alarm though command line.	~$ cat /home/profile ALARM=DISABLED PATH=/bin:/usr/bin:/home alarmon sleep 8	A message should appear after 5 seconds asking whether you want to kill the command. If user enters 'y' or 'Y' then current process should get killed and prompt should reappear. And if user enters 'n' or 'N' then shell should let the current command complete and then the prompt should appear.	~$ alarmon Enabling alarm ~$ sleep 8 Do you want to stop the program? (y/n) y Killing process ~$	**Pass**
3.	Disable alarm through command line.	~$ cat /home/profile ALARM=DISABLED PATH=/bin:/usr/bin:/home alarmoff	Alarm should get disabled and shell should let the command complete and then the prompt should appear.	~$ alarmoff Disabling alarm ~$ sleep 8 ~$	**Pass**

		sleep 8			
4.	Disable the alarm and kill the running process using Ctrl + c.	~$ cat /home/profile ALARM=DISABLED PATH=/bin:/usr/bin:/home alarmoff sleep 8 Ctrl + c	Alarm should get disabled and shell should let the command complete. The moment Ctrl + c is issued, the runnin command should get killed and shell prompt should reappear.	~$ alarmoff Disabling alarm ~$ sleep 8 ^C ~$	**Pass**

Conditional Command Execution using if-then-else

No.	Description	Input	Expected Output	Actual Output	Result
1.	Specifying multiple commands using if-then-else with a valid syntax.	if date ; then ls ;else pwd;fi;	date should get executed and if date is successful then ls should get executed else pwd should get executed.	~$ if date ; then ls ;else pwd;fi; Sun Sep 28 15:26:21 GMT 2014 ALIAS_PROFILE ast bin dateop.out dhanashree profile ~$	**Pass**
2.	Specifying multiple commands using if-then-else with a valid syntax.	if invalid ; then ls ;else pwd;fi;	invalid is not a valid command hence the pwd should get executed and not ls.	~$ if invalid; then ls; else pwd; fi; invalid Command not found in PATH. /home ~$	**Pass**
3.	Specifying multiple commands using invalid if-then-else syntax.	if date;then ls;else pwd	There is ; missing after else part and there is no ending fi hence syntax	~$ if date;then ls;else pwd Syntax error!	**Pass**

			error should be reported.	~$	
4.	Specifying multiple commands using invalid if-then-else syntax.	if date ;then ;ls;else pwd fi;	There is invalid "ls;" part which is not accompanied by either if, then or else hence syntax error should be reported.	~$ if date ;then ;ls;else pwd fi; Syntax error! ~$	**Pass**
5.	Specifying invalid commands resembling if.	fi date	Command not found error message should be reported as this is not a valid command.	~$ fi date fi Command not found in PATH. ~$	**Pass**