Mandatory Parts

Simple Command

- Execute a simple command with an absolute path like /bin/ls or any other command without options

Arguments

- Execute a simple command with an absolute path like /bin/ls or any other command with arguments but without quote and double quotes
- Repeat multiple times with different commands and arguments

echo

- Execute the echo command with or without arguments or options
- Repeat multiple times with different arguments

exit

- Execute exit command with or without arguments
- Repeat multiple times with different arguments
- Don’t forget to relaunch the minishell

Return value of a process

- Execute a simple command with absolute path like /bin/ls or any other command with arguments but without quotes and couple quotes then execute echo $?
- check the printed value. You can repeat the same in bash and compare it
- Repeat multiple times with different commands and arguments, use some failing commands like ‘/bin/ls filethatdoesntexist’

Semicolons

- Execute multiple simple commands with absolute path with arguments but separate them with semicolons
- Repeat multiple times with different commands and don’t forget to try with or without whitespaces around the semicolons

Signals

- ctrl-c in an empty prompt
- ctrl-\ in an empty prompt
- ctrl-d in an empty prompt
- ctrl-c in a prompt after you wrote some stuff
- ctrl-\ in a prompt after you wrote some stuff
- ctrl-d in a prompt after you wrote some stuff
- ctrl-c in a prompt after a blocking command like cat or grep without arguments
- ctrl-\ in a prompt after a blocking command like cat or grep without arguments
- ctrl-d in a prompt after a blocking command like cat or grep without arguments
- Repeat multiple times with different commands

Double Quotes

- Execute a simple command with absolute path with arguments but this time double quotes (you should include whitespaces and semicolons in the quotes)
- Think about empty arguments or a weird use of ‘\’
- Do not try multiline strings

env

- Check if env shows you the current environments variables

export

- export environment variables, create new ones and replace old ones
- Check them with env

unset

- Export environment variables, create new ones and replace old ones
- Use unset to remove some of them
- Check the result with env

Environment Variables

- Execute echo with some $ variables as arguments
- Check if double quotes around $ variables is working correctly (like in bash)

cd

- Use the command cd to move the working directory and check if you are in the right directory with /bin/ls
- Repeat multiple times with working and not working cd
- try ‘,’ ‘..’ as arguments too

pwd

- Use the command pwd
- Repeat multiple times in multiple directories

Relative Path

- Execute commands but this time use a relative path
- Repeat multiple times in multiple directories with complex relative path

Environment Path

- Execute commands but this time without any path (ls, wc, awk, etc…)
- Unset the $PATH and check if it is not working anymore
- Set the $PATH to a multiple directory value (directory1:directory2) and check that directories are checked in order from left to right

Simple Quotes

- Execute commands with simple quotes as argument
- Try empty arguments
- Try environment variables, whitespaces and semicolons in the simple quotes

Redirection

- Execute commands with redirections <and/or>
- Repeat multiple times with different commands and arguments and sometimes change > with >>
- Check if multiple of the same redirections fail

Pipes

- Execute commands with pipes like ‘cat file | grep bla | more’
- Repeat multiple times with different commands and arguments
- Try some failing commands like ‘ls filethatdoesntexist | grep bla | more’
- Try to mix pipes and redirections

Go Crazy

- Execute commands that should not work like ‘dsbksdgbksdghsd’ and check if the shell doesn’t crash and prints an error
- Try to execute a really really really long command with a ton of arguments
- Have fun with that beautiful minishell and enjoy it

Bonus

double left redirection (1)

- Check if << is working fine

Line editing (0 - 5)

- Can we move the cursor left and right and edit the line by inserting or deleting characters at cursor location
- Can we navigate through history with up and down
- Can we copy paste all/part of a line using a key sequence
- Can we move word by word with ctrl+left or ctrl+right
- Go directly to the beginning or the end of the line with home or end
- Write and edit commands with mutilines

And, Or (0 - 5)

- Use &&, || and parenthesis with commands and check if it works like bash
- For each working flag give 1 point
- if all flags are working give 1 bonus points

WildCard (1)

- Use wildcards in arguments
- Try things like */*
- Go crazy with wildcards