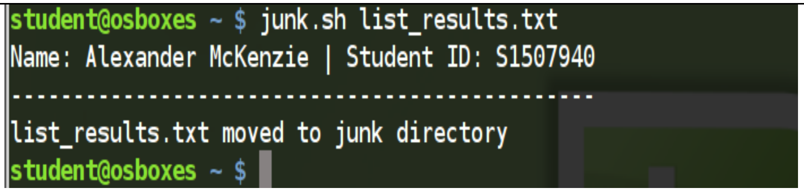
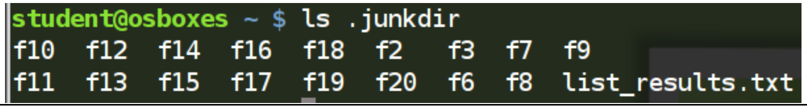
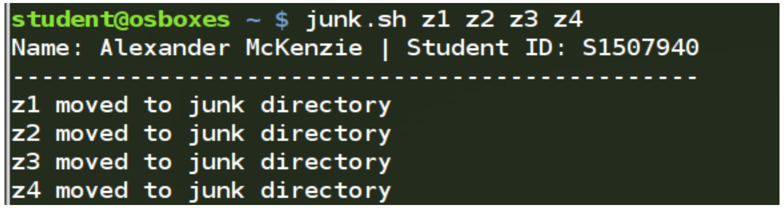
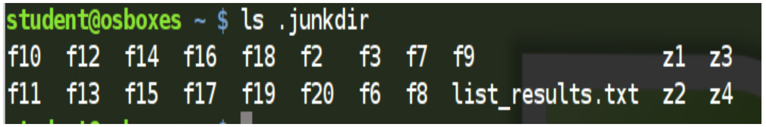
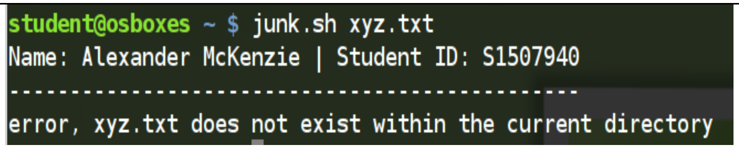
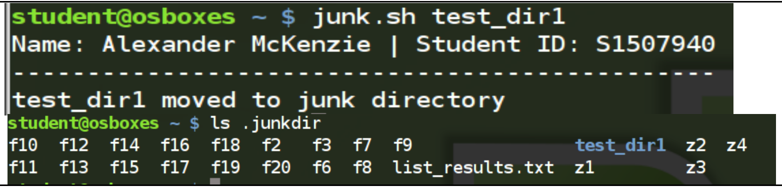
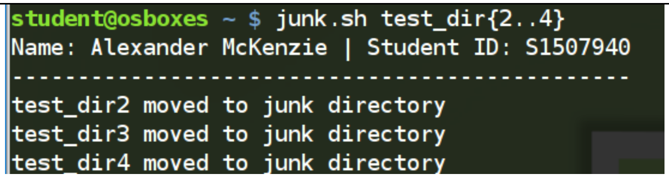
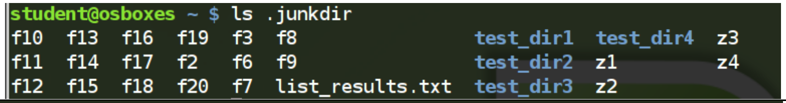
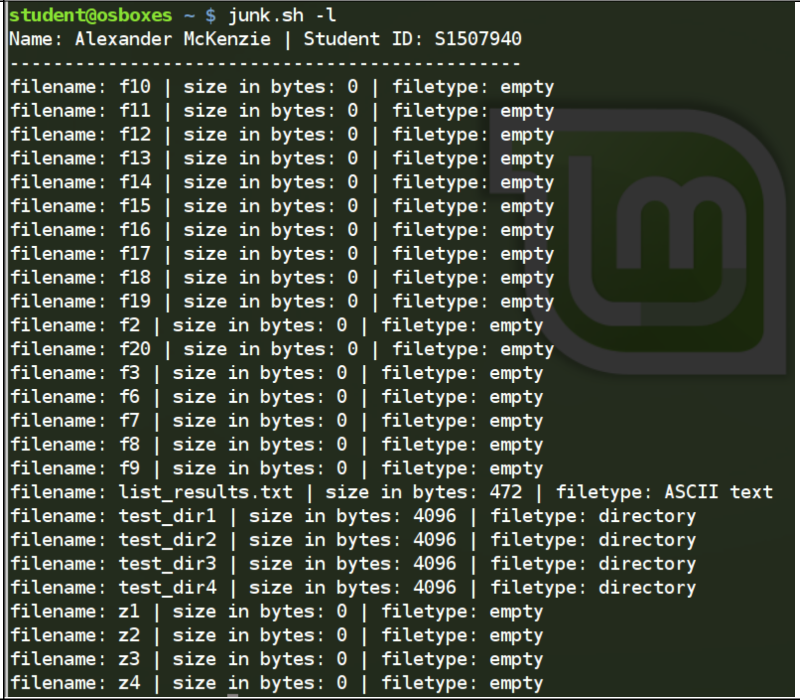
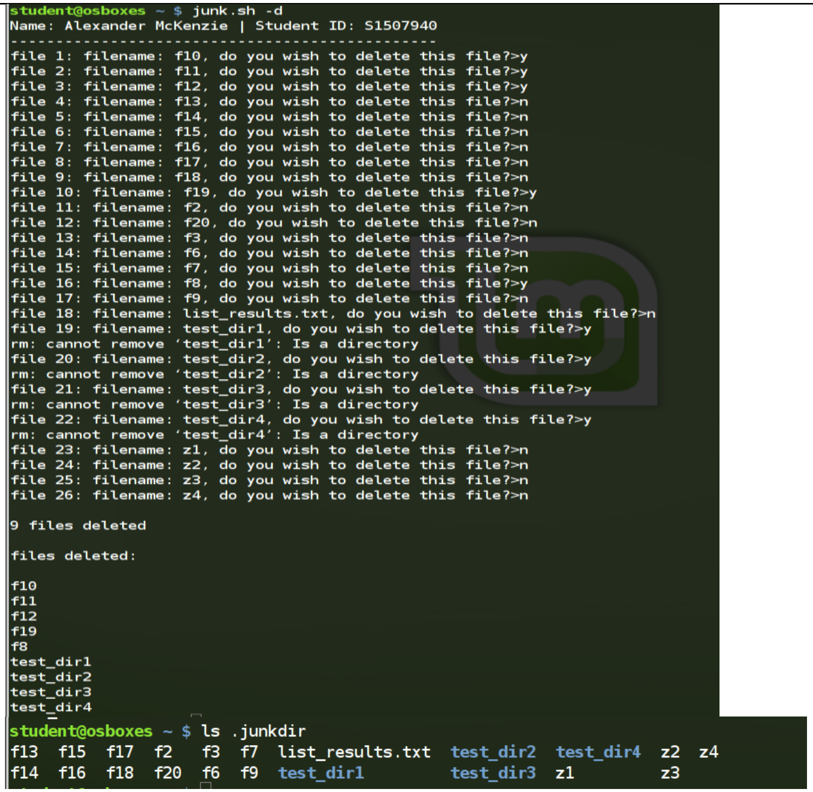
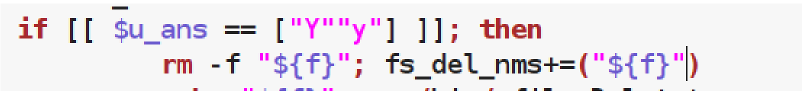
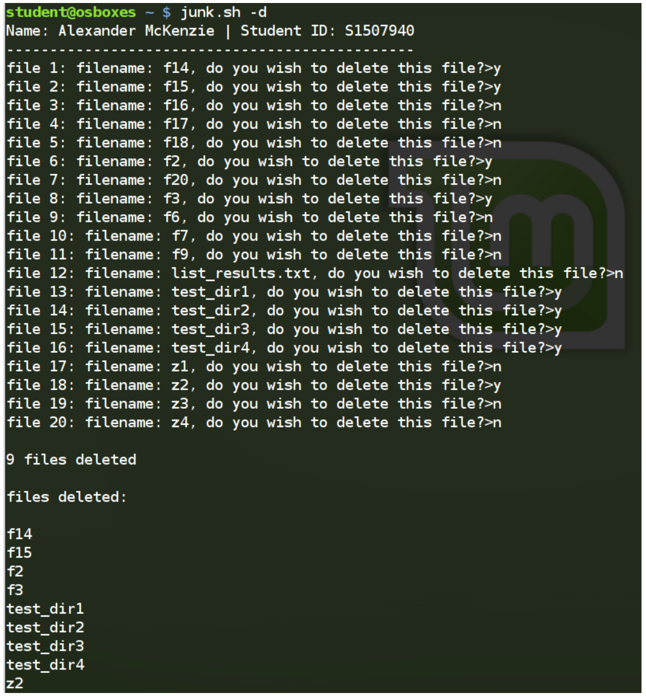
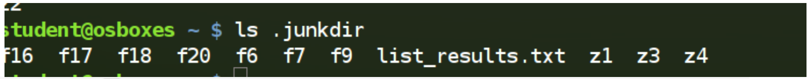
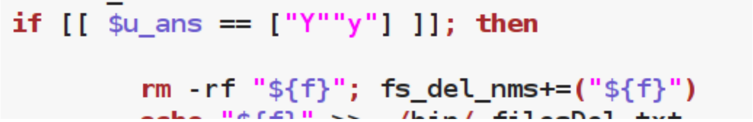
Test	Test Type	Expected result	Actual result Description	Actual result screenshots
Moving a file to .junkdir through junk shell script	Normal	The junk script should move the specified file to the .junkdir and confirm that it has done this through output to the terminal	The junk script moved the specified file to the .junkdir directory. this is shown through listing the contents of .junkdir.	 student@osboxes ~ $ junk.sh list_results.txt Name: Alexander McKenzie | Student ID: S1507940 --- list_results.txt moved to junk directory student@osboxes ~ $  student@osboxes ~ $ ls .junkdir f10 f12 f14 f16 f18 f2 f3 f7 f9 f11 f13 f15 f17 f19 f20 f6 f8 list_results.txt
Moving multiple files to .junkdir through junk shell script	Normal	The junk script should move the specified files to the .junkdir directory and confirm it has done this through output to the terminal	The junk script moved the specified files to the .junkdir directory. this is shown through listing the contents of .junkdir	 student@osboxes ~ $ junk.sh z1 z2 z3 z4 Name: Alexander McKenzie | Student ID: S1507940 --- z1 moved to junk directory z2 moved to junk directory z3 moved to junk directory z4 moved to junk directory  student@osboxes ~ $ ls .junkdir f10 f12 f14 f16 f18 f2 f3 f7 f9 z1 z3 f11 f13 f15 f17 f19 f20 f6 f8 list_results.txt z2 z4
Moving a non existent file to .junkdir through junk shell script	Exceptional	The junk script should output an error message	The junk script output an error message explaining why it could not move xyz.txt to the .junkdir directory	 student@osboxes ~ $ junk.sh xyz.txt Name: Alexander McKenzie | Student ID: S1507940 --- error, xyz.txt does not exist within the current directory
Moving a directory	Normal	The junk script should move the specified directory to the junk directory	The junk script moved the specified directory to the .junkdir directory, this is shown through listing the contents of .junkdir	 student@osboxes ~ $ junk.sh test_dir1 Name: Alexander McKenzie | Student ID: S1507940 --- test_dir1 moved to junk directory student@osboxes ~ $ ls .junkdir f10 f12 f14 f16 f18 f2 f3 f7 f9 test_dir1 z2 z4 f11 f13 f15 f17 f19 f20 f6 f8 list_results.txt z1 z3

Moving multiple directories	Normal	The junk script should move the specified directories to the .junkdir directory	The junk script moved the specified directories to the .junkdir, this is shown through listing the contents of .junkdir	 student@osboxes ~ $ junk.sh test_dir{2..4} Name: Alexander McKenzie | Student ID: S1507940 -- test_dir2 moved to junk directory test_dir3 moved to junk directory test_dir4 moved to junk directory  student@osboxes ~ $ ls .junkdir f10 f13 f16 f19 f3 f8 test_dir1 test_dir4 z3 f11 f14 f17 f2 f6 f9 test_dir2 z1 z4 f12 f15 f18 f20 f7 list_results.txt test_dir3 z2
Listing files within .junkdir	Normal	The junk script should list all files within the .junkdir alongside their size in bytes and their filetype	The junk script listed all files within the .junkdir alongside their size in bytes and their filetype	 student@osboxes ~ $ junk.sh -l Name: Alexander McKenzie | Student ID: S1507940 -- filename: f10 | size in bytes: 0 | filetype: empty filename: f11 | size in bytes: 0 | filetype: empty filename: f12 | size in bytes: 0 | filetype: empty filename: f13 | size in bytes: 0 | filetype: empty filename: f14 | size in bytes: 0 | filetype: empty filename: f15 | size in bytes: 0 | filetype: empty filename: f16 | size in bytes: 0 | filetype: empty filename: f17 | size in bytes: 0 | filetype: empty filename: f18 | size in bytes: 0 | filetype: empty filename: f19 | size in bytes: 0 | filetype: empty filename: f2 | size in bytes: 0 | filetype: empty filename: f20 | size in bytes: 0 | filetype: empty filename: f3 | size in bytes: 0 | filetype: empty filename: f6 | size in bytes: 0 | filetype: empty filename: f7 | size in bytes: 0 | filetype: empty filename: f8 | size in bytes: 0 | filetype: empty filename: f9 | size in bytes: 0 | filetype: empty filename: list_results.txt | size in bytes: 472 | filetype: ASCII text filename: test_dir1 | size in bytes: 4096 | filetype: directory filename: test_dir2 | size in bytes: 4096 | filetype: directory filename: test_dir3 | size in bytes: 4096 | filetype: directory filename: test_dir4 | size in bytes: 4096 | filetype: directory filename: z1 | size in bytes: 0 | filetype: empty filename: z2 | size in bytes: 0 | filetype: empty filename: z3 | size in bytes: 0 | filetype: empty filename: z4 | size in bytes: 0 | filetype: empty

Deleting files and directories within .junkdir (valid inputs)	Normal	The junk script should delete the specified file or directory when the user inputs y or ignore the file if the user inputs n. The junk script should then output the files deleted	The junk script could delete regular files however it was not able to delete directories as it attempted to use the rm -f command for file removal regardless of file type. Evidence of regular files being correctly removed is shown through listing contents of the .junkdir directory	 

```
student@osboxes ~ $ junk.sh -d
Name: Alexander McKenzie | Student ID: S1507940
------------------------------------------------
file 1: filename: f10, do you wish to delete this file?>y
file 2: filename: f11, do you wish to delete this file?>y
file 3: filename: f12, do you wish to delete this file?>y
file 4: filename: f13, do you wish to delete this file?>n
file 5: filename: f14, do you wish to delete this file?>n
file 6: filename: f15, do you wish to delete this file?>n
file 7: filename: f16, do you wish to delete this file?>n
file 8: filename: f17, do you wish to delete this file?>n
file 9: filename: f18, do you wish to delete this file?>n
file 10: filename: f19, do you wish to delete this file?>y
file 11: filename: f2, do you wish to delete this file?>n
file 12: filename: f20, do you wish to delete this file?>n
file 13: filename: f3, do you wish to delete this file?>n
file 14: filename: f6, do you wish to delete this file?>n
file 15: filename: f7, do you wish to delete this file?>n
file 16: filename: f8, do you wish to delete this file?>y
file 17: filename: f9, do you wish to delete this file?>n
file 18: filename: list_results.txt, do you wish to delete this file?>n
file 19: filename: test_dir1, do you wish to delete this file?>y
rm: cannot remove ‘test_dir1’: Is a directory
file 20: filename: test_dir2, do you wish to delete this file?>y
rm: cannot remove ‘test_dir2’: Is a directory
file 21: filename: test_dir3, do you wish to delete this file?>y
rm: cannot remove ‘test_dir3’: Is a directory
file 22: filename: test_dir4, do you wish to delete this file?>y
rm: cannot remove ‘test_dir4’: Is a directory
file 23: filename: z1, do you wish to delete this file?>n
file 24: filename: z2, do you wish to delete this file?>n
file 25: filename: z3, do you wish to delete this file?>n
file 26: filename: z4, do you wish to delete this file?>n

9 files deleted

files deleted:

f10
f11
f12
f19
f8
test_dir1
test_dir2
test_dir3
test_dir4
student@osboxes ~ $ ls .junkdir
f13  f15  f17  f2   f3  f7  list_results.txt  test_dir2  test_dir4  z2  z4
f14  f16  f18  f20  f6  f9  test_dir1         test_dir3  z1         z3
```

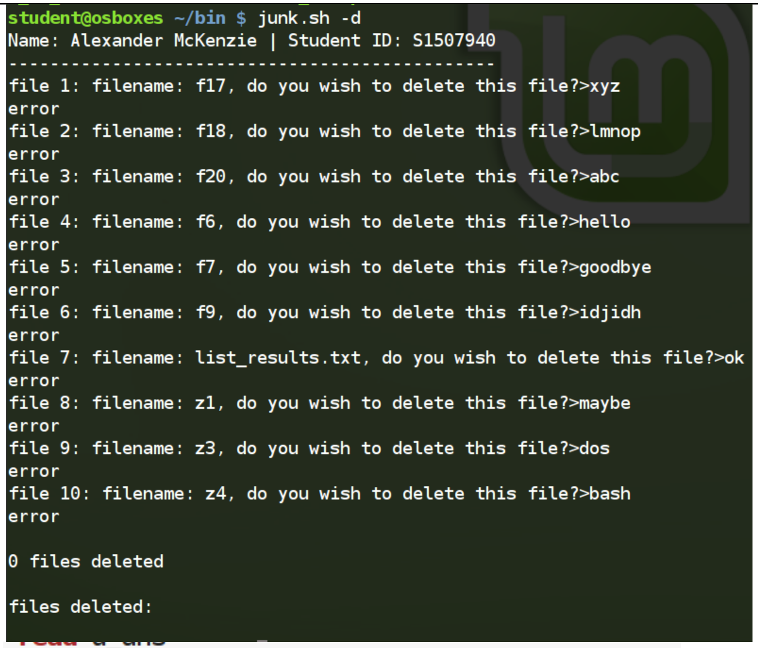
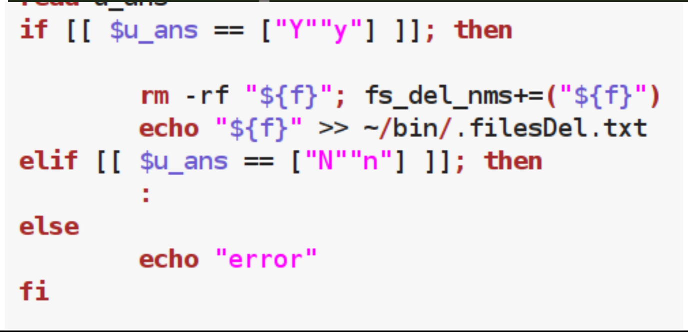
```
if [[ $u_ans == ["Y""y"] ]]; then
        rm -f "${f}"; fs_del_nms+=("${f}")
```

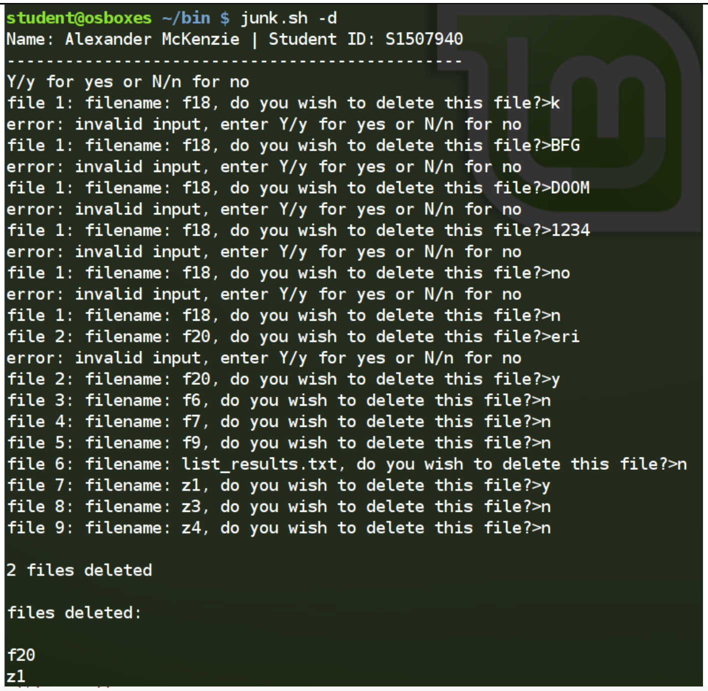
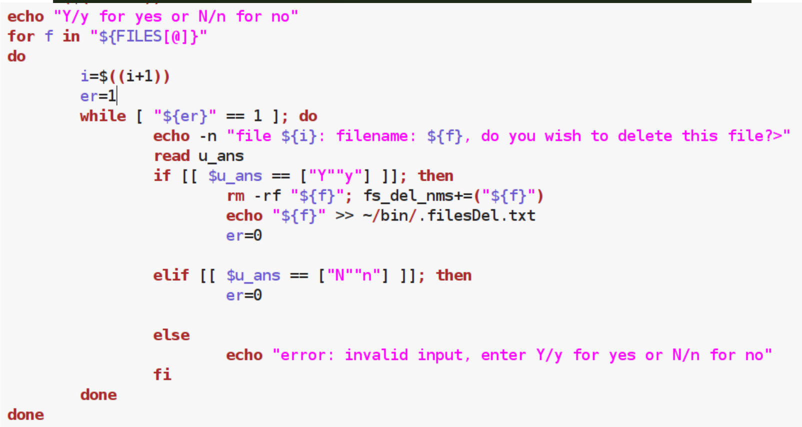
Deleting files and directories within .junkdir (valid inputs & after debug)	Normal	The junk script should delete the specified file or directory when the user inputs y or ignore the file if the user inputs n. The junk script should then output the files deleted	After changing the rm -f command used to delete files to rm -rf (-r allowing for deletion of directories and their contents) the junk script delete function could delete regular files and directories.	  

```
student@osboxes ~ $ junk.sh -d
Name: Alexander McKenzie | Student ID: S1507940
----------------------------------------------
file 1: filename: f14, do you wish to delete this file?>y
file 2: filename: f15, do you wish to delete this file?>y
file 3: filename: f16, do you wish to delete this file?>n
file 4: filename: f17, do you wish to delete this file?>n
file 5: filename: f18, do you wish to delete this file?>n
file 6: filename: f2, do you wish to delete this file?>y
file 7: filename: f20, do you wish to delete this file?>n
file 8: filename: f3, do you wish to delete this file?>y
file 9: filename: f6, do you wish to delete this file?>n
file 10: filename: f7, do you wish to delete this file?>n
file 11: filename: f9, do you wish to delete this file?>n
file 12: filename: list_results.txt, do you wish to delete this file?>n
file 13: filename: test_dir1, do you wish to delete this file?>y
file 14: filename: test_dir2, do you wish to delete this file?>y
file 15: filename: test_dir3, do you wish to delete this file?>y
file 16: filename: test_dir4, do you wish to delete this file?>y
file 17: filename: z1, do you wish to delete this file?>n
file 18: filename: z2, do you wish to delete this file?>y
file 19: filename: z3, do you wish to delete this file?>n
file 20: filename: z4, do you wish to delete this file?>n

9 files deleted

files deleted:

f14
f15
f2
f3
test_dir1
test_dir2
test_dir3
test_dir4
z2
```

```
z2
student@osboxes ~ $ ls .junkdir
f16  f17  f18  f20  f6  f7  f9  list_results.txt  z1  z3  z4
```

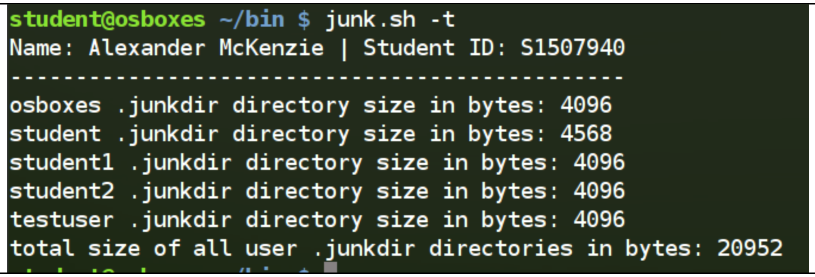
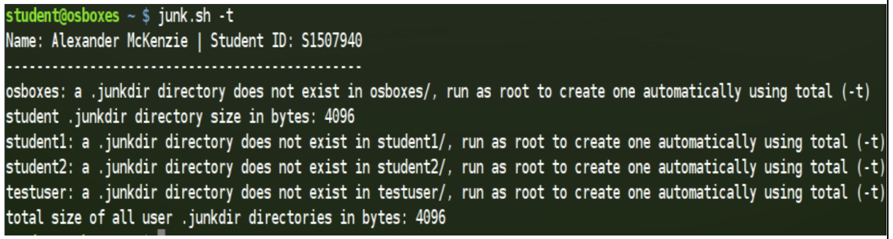
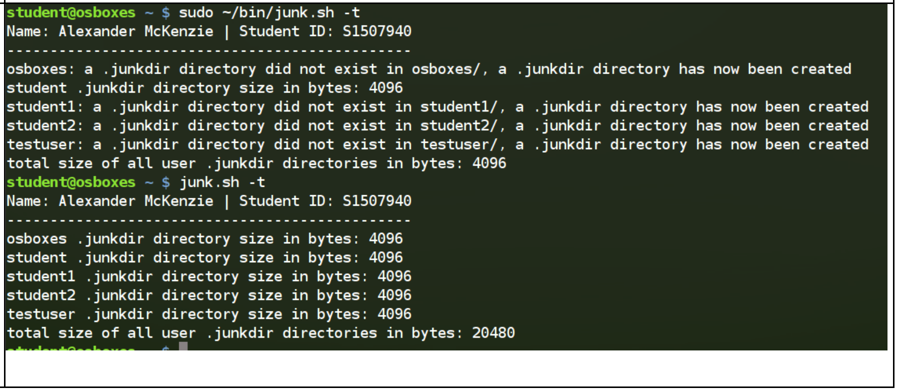
```
if [[ $u_ans == ["Y""y"] ]]; then

        rm -rf "${f}"; fs_del_nms+=("${f}")
        echo "${f}" >> ~/bin/.filesDel.txt
```

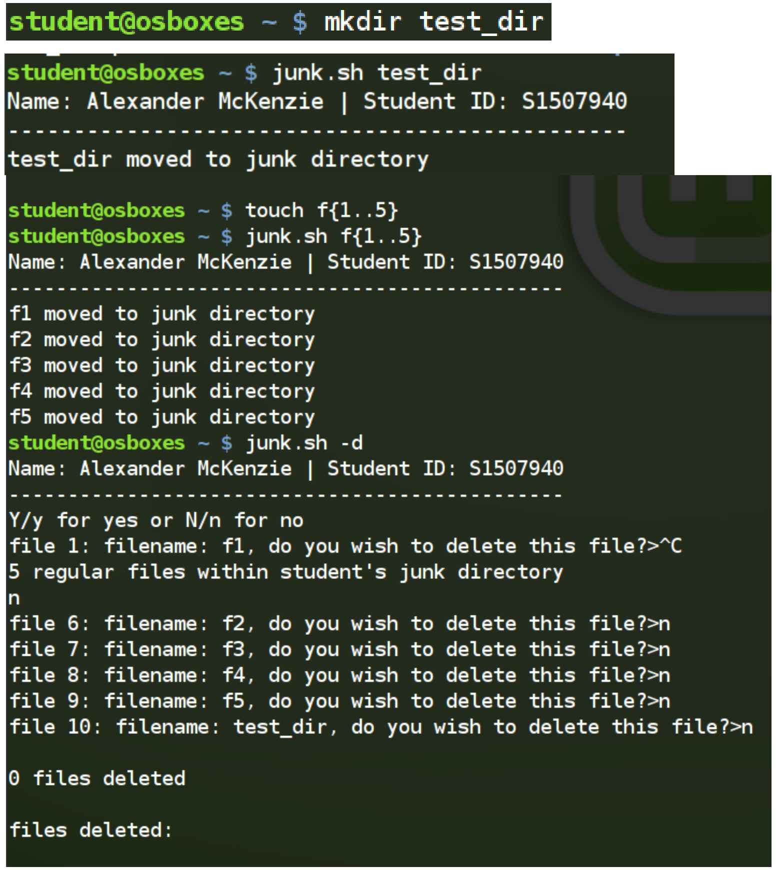
Deleting files and directories within .junkdir (invalid inputs)	Exceptional	The junk script should output that an error has occurred and correctly handle this error	The junk script outputs that an error has occurred however it doesn’t handle the error very effectively, this is due to code design oversight.	
				`student@osboxes ~/bin $ junk.sh -d`
				`Name: Alexander McKenzie | Student ID: S1507940`
				`---------------------------------------------`
				`file 1: filename: f17, do you wish to delete this file?>xyz`
				`error`
				`file 2: filename: f18, do you wish to delete this file?>lmnop`
				`error`
				`file 3: filename: f20, do you wish to delete this file?>abc`
				`error`
				`file 4: filename: f6, do you wish to delete this file?>hello`
				`error`
				`file 5: filename: f7, do you wish to delete this file?>goodbye`
				`error`
				`file 6: filename: f9, do you wish to delete this file?>idjidh`
				`error`
				`file 7: filename: list_results.txt, do you wish to delete this file?>ok`
				`error`
				`file 8: filename: z1, do you wish to delete this file?>maybe`
				`error`
				`file 9: filename: z3, do you wish to delete this file?>dos`
				`error`
				`file 10: filename: z4, do you wish to delete this file?>bash`
				`error`
				
				`0 files deleted`
				
				`files deleted:`
				
				`if [[ $u_ans == ["Y""y"] ]]; then`
				
				`        rm -rf "${f}"; fs_del_nms+=("${f}")`
				`        echo "${f}" >> ~/bin/.filesDel.txt`
				`elif [[ $u_ans == ["N""n"] ]]; then`
				`        :`
				`else`
				`        echo "error"`
				`fi`

Deleting files and directories within .junkdir (invalid inputs & after code re-design)	Exceptional	The junk script should output that an error has occurred and correctly handle this error	The junk script now correctly handles errors. While improving error handling I also noticed that the script did not provide much user feedback as to how users should enter inputs, I decided to add feedback as to how the user should respond.	 

```
student@osboxes ~/bin $ junk.sh -d
Name: Alexander McKenzie | Student ID: S1507940
----------------------------------------------
Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>k
error: invalid input, enter Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>BFG
error: invalid input, enter Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>DOOM
error: invalid input, enter Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>1234
error: invalid input, enter Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>no
error: invalid input, enter Y/y for yes or N/n for no
file 1: filename: f18, do you wish to delete this file?>n
file 2: filename: f20, do you wish to delete this file?>eri
error: invalid input, enter Y/y for yes or N/n for no
file 2: filename: f20, do you wish to delete this file?>y
file 3: filename: f6, do you wish to delete this file?>n
file 4: filename: f7, do you wish to delete this file?>n
file 5: filename: f9, do you wish to delete this file?>n
file 6: filename: list_results.txt, do you wish to delete this file?>n
file 7: filename: z1, do you wish to delete this file?>y
file 8: filename: z3, do you wish to delete this file?>n
file 9: filename: z4, do you wish to delete this file?>n

2 files deleted

files deleted:

f20
z1
```

```
echo "Y/y for yes or N/n for no"
for f in "${FILES[@]}"
do
        i=$((i+1))
        er=1
        while [ "${er}" == 1 ]; do
                echo -n "file ${i}: filename: ${f}, do you wish to delete this file?>"
                read u_ans
                if [[ $u_ans == ["Y""y"] ]]; then
                        rm -rf "${f}"; fs_del_nms+=("${f}")
                        echo "${f}" >> ~/bin/.filesDel.txt
                        er=0

                elif [[ $u_ans == ["N""n"] ]]; then
                        er=0

                else
                        echo "error: invalid input, enter Y/y for yes or N/n for no"
                fi
        done
done
```

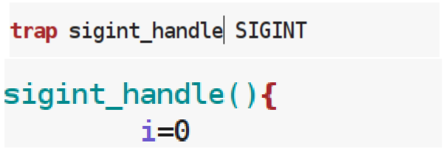
Getting total size of all user .junkdir directories	Normal	The junk script should list the size in bytes of all user junk directories then display the total size in bytes of all user junk directories	The junk script listed the size in bytes of all user junk directories then displayed the total size in bytes of all user junk directories	 student@osboxes ~/bin $ junk.sh -t Name: Alexander McKenzie | Student ID: S1507940 -- osboxes .junkdir directory size in bytes: 4096 student .junkdir directory size in bytes: 4568 student1 .junkdir directory size in bytes: 4096 student2 .junkdir directory size in bytes: 4096 testuser .junkdir directory size in bytes: 4096 total size of all user .junkdir directories in bytes: 20952
Getting total size of all user .junkdir directories when no .junkdir directories exist (run as non-super user)	Normal	The junk script should tell the user that a .junkdir directory does not exist for each user and tell the user to run as super user in order to auto create these directories; excluding the current user, which it should make a directory for automatically through regular running of the junk script (i.e. outwith the total function).	The junk script tells the user to run as root in order to automatically generate .junkdir directories for other users	 student@osboxes ~ $ junk.sh -t Name: Alexander McKenzie | Student ID: S1507940 -- osboxes: a .junkdir directory does not exist in osboxes/, run as root to create one automatically using total (-t) student .junkdir directory size in bytes: 4096 student1: a .junkdir directory does not exist in student1/, run as root to create one automatically using total (-t) student2: a .junkdir directory does not exist in student2/, run as root to create one automatically using total (-t) testuser: a .junkdir directory does not exist in testuser/, run as root to create one automatically using total (-t) total size of all user .junkdir directories in bytes: 4096
Getting total size of all user .junkdir directories when no .junkdir directories exist (run as super user)	Normal	The junk script should automatically create a junkdir directory for each user, get the total size for each .junkdir directory and output the total size for all .junkdir directories	The junk script tells the user that junk directories do not exist in certain user directories where applicable and creates these directories, this is shown by running junk total again	 student@osboxes ~ $ sudo ~/bin/junk.sh -t Name: Alexander McKenzie | Student ID: S1507940 -- osboxes: a .junkdir directory did not exist in osboxes/, a .junkdir directory has now been created student .junkdir directory size in bytes: 4096 student1: a .junkdir directory did not exist in student1/, a .junkdir directory has now been created student2: a .junkdir directory did not exist in student2/, a .junkdir directory has now been created testuser: a .junkdir directory did not exist in testuser/, a .junkdir directory has now been created total size of all user .junkdir directories in bytes: 4096 student@osboxes ~ $ junk.sh -t Name: Alexander McKenzie | Student ID: S1507940 -- osboxes .junkdir directory size in bytes: 4096 student .junkdir directory size in bytes: 4096 student1 .junkdir directory size in bytes: 4096 student2 .junkdir directory size in bytes: 4096 testuser .junkdir directory size in bytes: 4096 total size of all user .junkdir directories in bytes: 20480

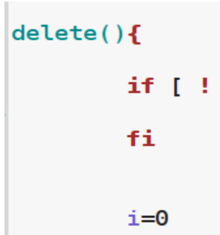
<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr>
<td>Carrying out a SIGINT interrupt on the junk script</td>
<td>Normal</td>
<td>The junk script should display the total amount of regular files within the users .junkdir directory</td>
<td>The junk script was able to show the amount of regular files within the current users .junkdir (demonstrated by the count ignoring the directory test_dir present within the .junkdir directory) however the count used for counting the amount of files processed increased by 5 in the original delete function, I believed this was due to using the same variable name for the counter variable in both the sigint_handle function and the delete function</td>
<td>
<pre>
student@osboxes ~ $ mkdir test_dir
student@osboxes ~ $ junk.sh test_dir
Name: Alexander McKenzie | Student ID: S1507940
----
test_dir moved to junk directory

student@osboxes ~ $ touch f{1..5}
student@osboxes ~ $ junk.sh f{1..5}
Name: Alexander McKenzie | Student ID: S1507940
----
f1 moved to junk directory
f2 moved to junk directory
f3 moved to junk directory
f4 moved to junk directory
f5 moved to junk directory
student@osboxes ~ $ junk.sh -d
Name: Alexander McKenzie | Student ID: S1507940

Y/y for yes or N/n for no
file 1: filename: f1, do you wish to delete this file?>^C
5 regular files within student's junk directory
n
file 6: filename: f2, do you wish to delete this file?>n
file 7: filename: f3, do you wish to delete this file?>n
file 8: filename: f4, do you wish to delete this file?>n
file 9: filename: f5, do you wish to delete this file?>n
file 10: filename: test_dir, do you wish to delete this file?>n

0 files deleted

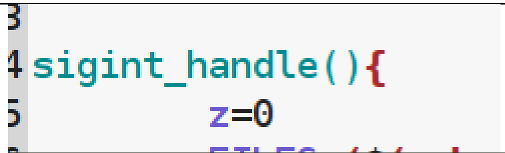
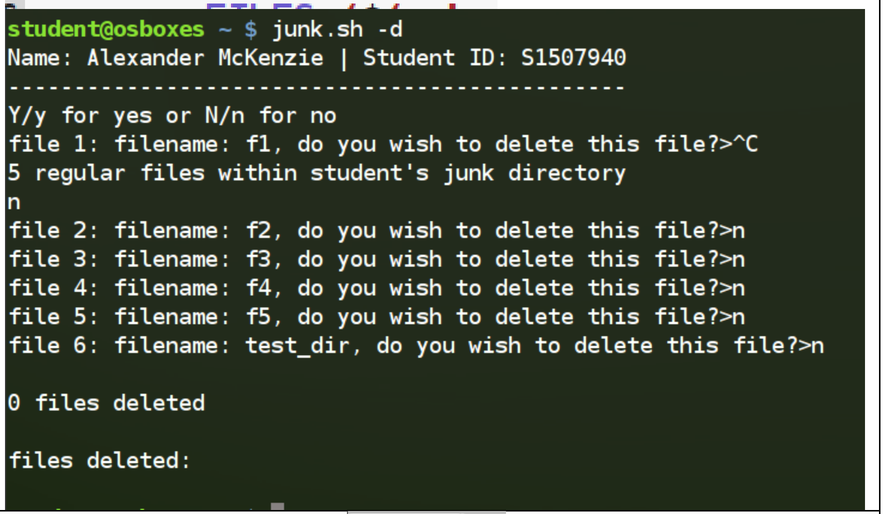
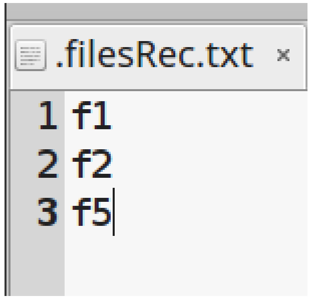
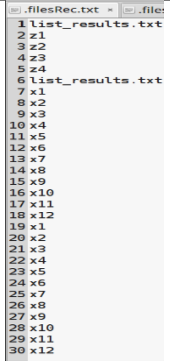
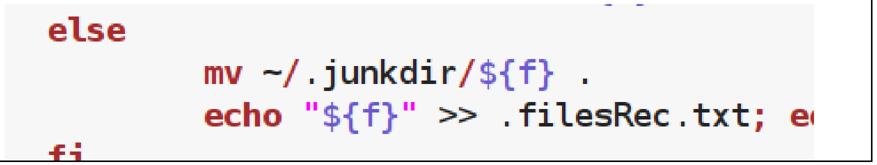
files deleted:
</pre>

<pre>
trap sigint_handle SIGINT

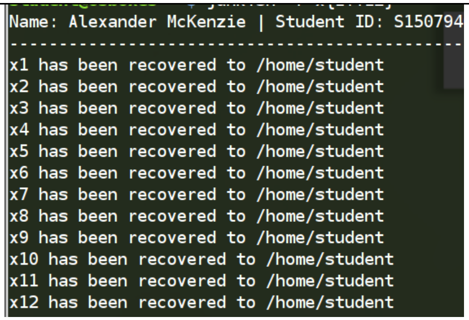
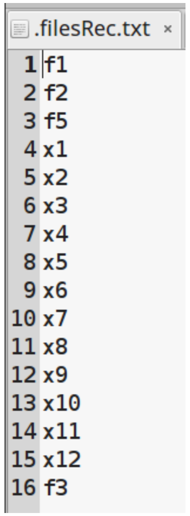
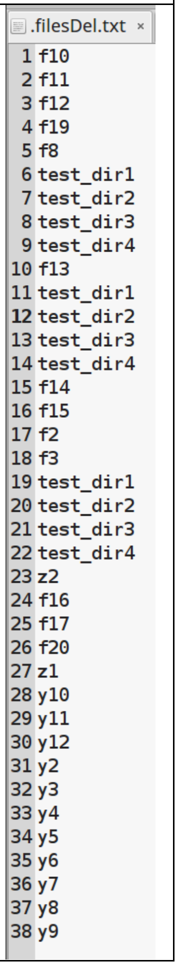
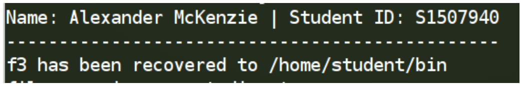
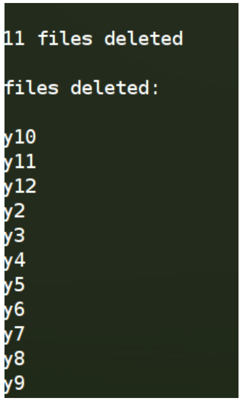
sigint_handle(){
        i=0
</pre>

<pre>
delete(){

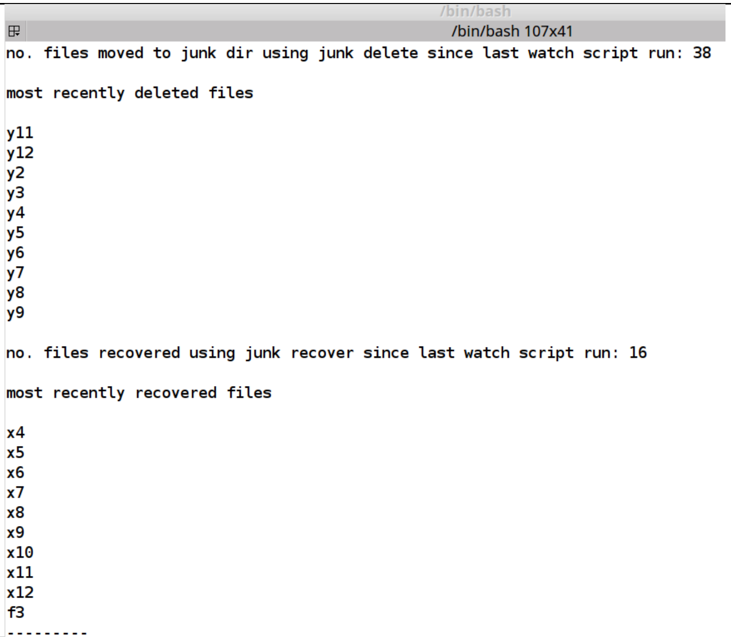
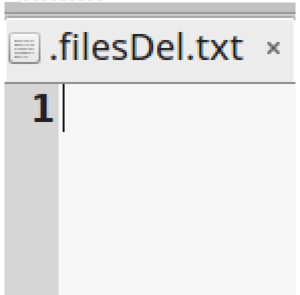
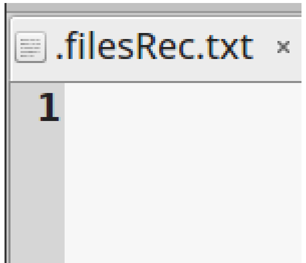
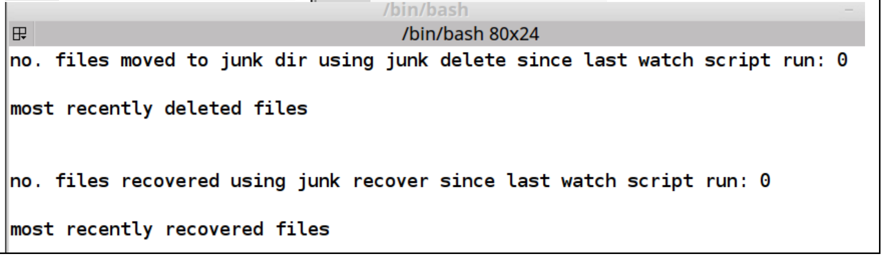
        if [!

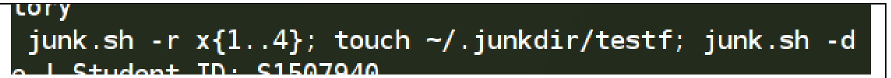
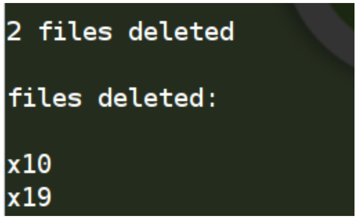
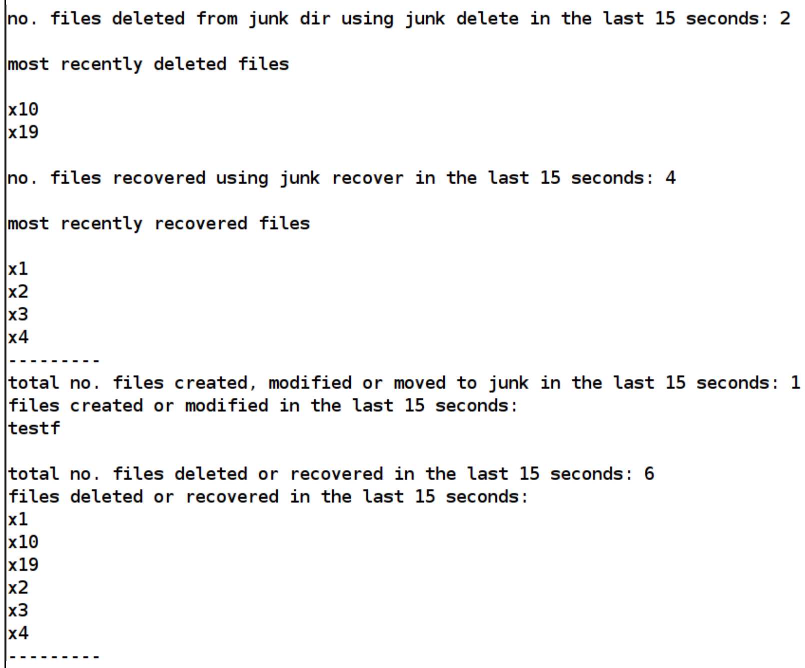
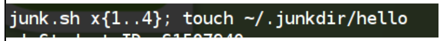
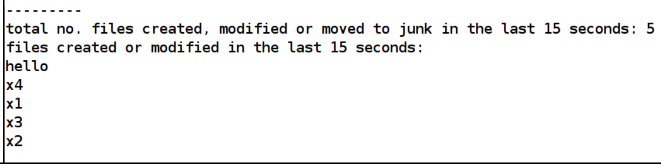
        fi

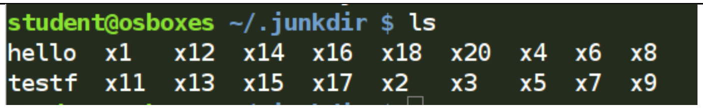
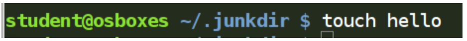
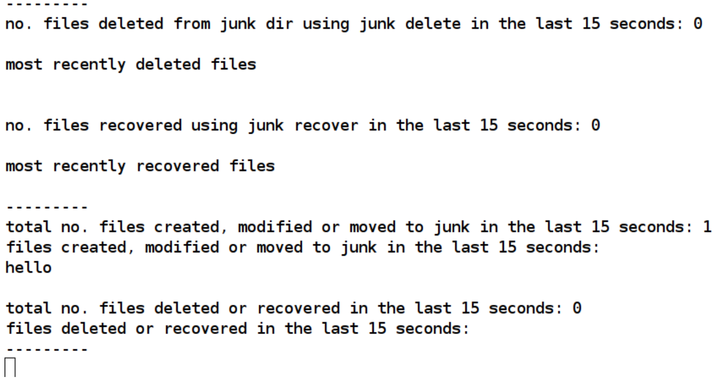
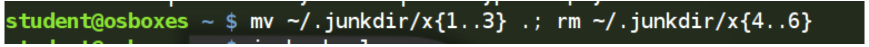
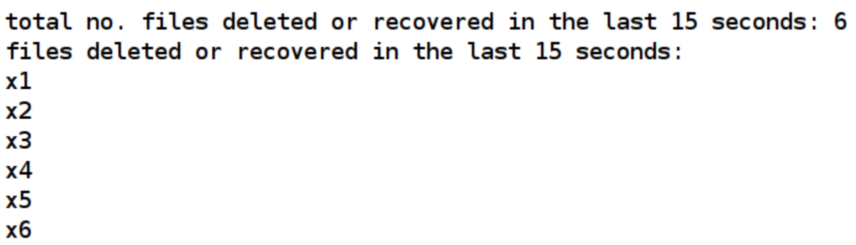
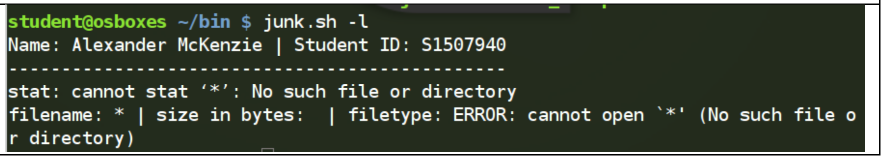
        i=0
</pre>
</td>
</tr>
</table>

Carrying out a SIGINT interrupt on the junk script(after debug)	Normal	The junk script should display the total amount of regular files within the users .junkdir directory without increasing the count for files processed in the delete function	After changing the counter variable used in sigint_handle from i to z the file count for each file processed in the delete function no longer increased by the number of regular files in the users junk directory.	 sigint_handle(){ z=0  student@osboxes ~ $ junk.sh -d Name: Alexander McKenzie | Student ID: S1507940 -- Y/y for yes or N/n for no file 1: filename: f1, do you wish to delete this file?>^C 5 regular files within student's junk directory n file 2: filename: f2, do you wish to delete this file?>n file 3: filename: f3, do you wish to delete this file?>n file 4: filename: f4, do you wish to delete this file?>n file 5: filename: f5, do you wish to delete this file?>n file 6: filename: test_dir, do you wish to delete this file?>n 0 files deleted files deleted:
Initial startup of watch script	Normal	The watch script should display the total amount of files recovered and files deleted since the last watch script run using the hidden text files stored in the bin directory used to store this information	Before running the watch script I noticed that when recovering files I had only specified the output to go to .filesRec.txt rather than a specified filepath, this meant that the .filesRec file would be stored in the current working directory rather than the correct bin directory. I debugged this issue before running the watch script.	.filesRec in bin  .filesRec.txt f1 f2 f5  .filesRec.txt list_results.txt z1 z2 z3 z4 list_results.txt x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 .filesRec in user home dir  else mv ~/.junkdir/${f} . echo "${f}" >> .filesRec.txt; e fi

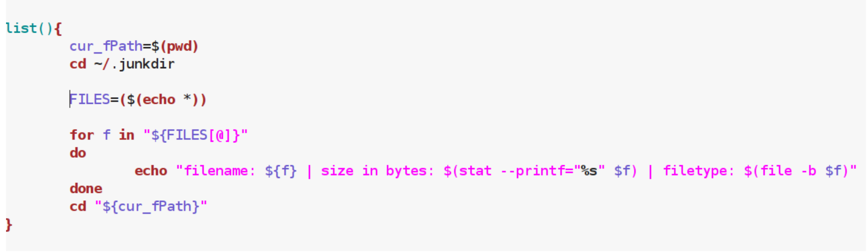
Initial startup of watch script (after debug)	Normal	The watch script should display the total amount of files recovered and deleted and 10 of the most recently recovered and deleted files since the last watch script run. The script should use the hidden text files stored in the bin directory used to store this information. The script should then wipe the information stored in the hidden text files.	The watch script displays the total amount of files recovered and deleted and 10 of the most recently recovered and deleted files since the last watch script run. The script also wipes the hidden text files after displaying this information.	 Name: Alexander McKenzie | Student ID: S150794 --- x1 has been recovered to /home/student x2 has been recovered to /home/student x3 has been recovered to /home/student x4 has been recovered to /home/student x5 has been recovered to /home/student x6 has been recovered to /home/student x7 has been recovered to /home/student x8 has been recovered to /home/student x9 has been recovered to /home/student x10 has been recovered to /home/student x11 has been recovered to /home/student x12 has been recovered to /home/student  .filesRec.txt × f1 f2 f5 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 f3  .filesDel.txt × f10 f11 f12 f19 f8 test_dir1 test_dir2 test_dir3 test_dir4 f13 test_dir1 test_dir2 test_dir3 test_dir4 f14 f15 f2 f3 test_dir1 test_dir2 test_dir3 test_dir4 z2 f16 f17 f20 z1 y10 y11 y12 y2 y3 y4 y5 y6 y7 y8 y9  Name: Alexander McKenzie | Student ID: S1507940 --- f3 has been recovered to /home/student/bin  11 files deleted files deleted: y10 y11 y12 y2 y3 y4 y5 y6 y7 y8 y9 Screenshots continued next page

				 no. files moved to junk dir using junk delete since last watch script run: 38 most recently deleted files y11 y12 y2 y3 y4 y5 y6 y7 y8 y9 no. files recovered using junk recover since last watch script run: 16 most recently recovered files x4 x5 x6 x7 x8 x9 x10 x11 x12 f3 ---------    no. files moved to junk dir using junk delete since last watch script run: 0 most recently deleted files no. files recovered using junk recover since last watch script run: 0 most recently recovered files

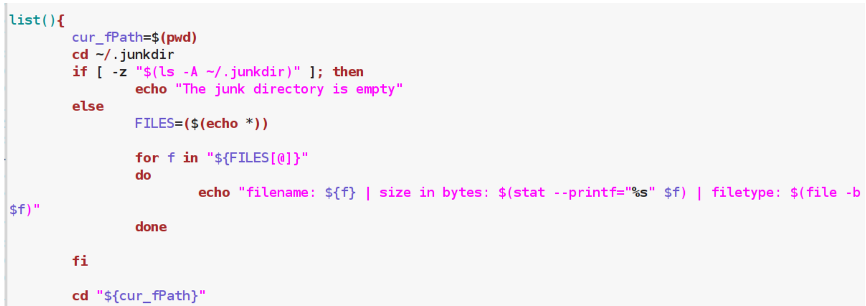
Watch script output	Normal	The watch script should display the files recovered and deleted through the use of junk delete and junk recover. The watch script should also display all files created, modified, moved to junk dir, deleted and recovered in the last 15 seconds.	The watch script was able to output all files recovered and deleted through the use of junk delete and junk recover. The watch script was also able to show all files created, modified, moved to junk dir, deleted and recovered in the last 15 seconds. I noticed that the output for files created or modified was incorrect as it should have also included files moved to the junk dir so I changed it’s output to reflect this.	 tory junk.sh -r x{1..4}; touch ~/.junkdir/testf; junk.sh -d e | Student ID: S1507940  2 files deleted files deleted: x10 x19  no. files deleted from junk dir using junk delete in the last 15 seconds: 2 most recently deleted files x10 x19 no. files recovered using junk recover in the last 15 seconds: 4 most recently recovered files x1 x2 x3 x4 --------- total no. files created, modified or moved to junk in the last 15 seconds: 1 files created or modified in the last 15 seconds: testf total no. files deleted or recovered in the last 15 seconds: 6 files deleted or recovered in the last 15 seconds: x1 x10 x19 x2 x3 x4 ---------  junk.sh x{1..4}; touch ~/.junkdir/hello  --------- total no. files created, modified or moved to junk in the last 15 seconds: 5 files created or modified in the last 15 seconds: hello x4 x1 x3 x2

				 student@osboxes ~/.junkdir $ ls hello x1 x12 x14 x16 x18 x20 x4 x6 x8 testf x11 x13 x15 x17 x2 x3 x5 x7 x9  student@osboxes ~/.junkdir $ touch hello  --------- no. files deleted from junk dir using junk delete in the last 15 seconds: 0 most recently deleted files no. files recovered using junk recover in the last 15 seconds: 0 most recently recovered files --------- total no. files created, modified or moved to junk in the last 15 seconds: 1 files created, modified or moved to junk in the last 15 seconds: hello total no. files deleted or recovered in the last 15 seconds: 0 files deleted or recovered in the last 15 seconds: ---------  student@osboxes ~ $ mv ~/.junkdir/x{1..3} .; rm ~/.junkdir/x{4..6}  total no. files deleted or recovered in the last 15 seconds: 6 files deleted or recovered in the last 15 seconds: x1 x2 x3 x4 x5 x6
Attempting to list an empty junk directory	Extreme	The junk script should output that the junk directory is empty	The junk script was not able to deal with an empty junk directory and displays incorrect output	 student@osboxes ~/bin $ junk.sh -l Name: Alexander McKenzie | Student ID: S1507940 --- stat: cannot stat '*': No such file or directory filename: * | size in bytes: | filetype: ERROR: cannot open `*' (No such file o r directory)

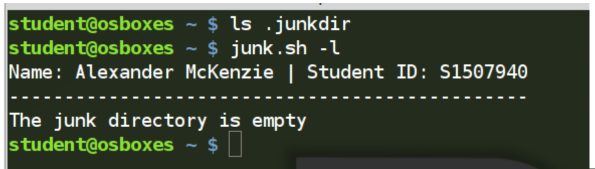
<table>
<tr>
<td>Attempting to list an empty junk directory (after code redesign)</td>
<td>Extreme</td>
<td>The junk script should output that the junk directory is empty</td>
<td>After a code edit the junk directory was able to correctly handle an empty junk directory.</td>
<td>



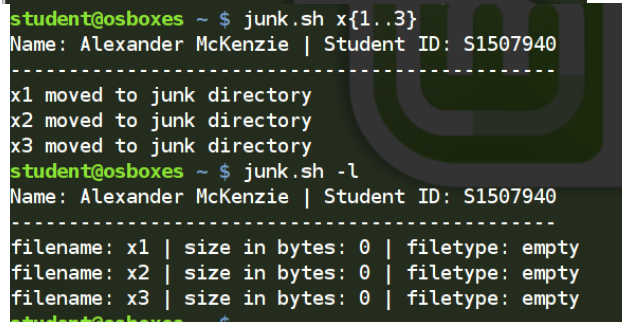
```
list(){
        cur_fPath=$(pwd)
        cd ~/.junkdir

        FILES=($(echo *))

        for f in "${FILES[@]}"
        do
                echo "filename: ${f} | size in bytes: $(stat --printf="%s" $f) | filetype: $(file -b $f)"
        done
        cd "${cur_fPath}"
}
```



```
list(){
        cur_fPath=$(pwd)
        cd ~/.junkdir
        if [ -z "$(ls -A ~/.junkdir)" ]; then
                echo "The junk directory is empty"
        else
                FILES=($(echo *))

                for f in "${FILES[@]}"
                do
                        echo "filename: ${f} | size in bytes: $(stat --printf="%s" $f) | filetype: $(file -b
$f)"
                done

        fi

        cd "${cur_fPath}"
```

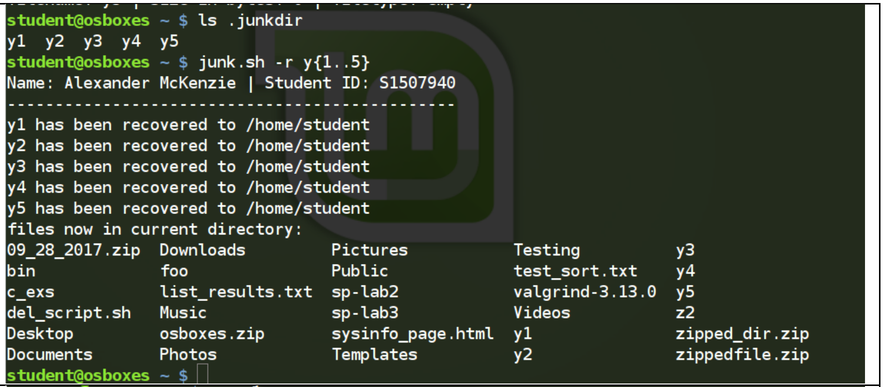
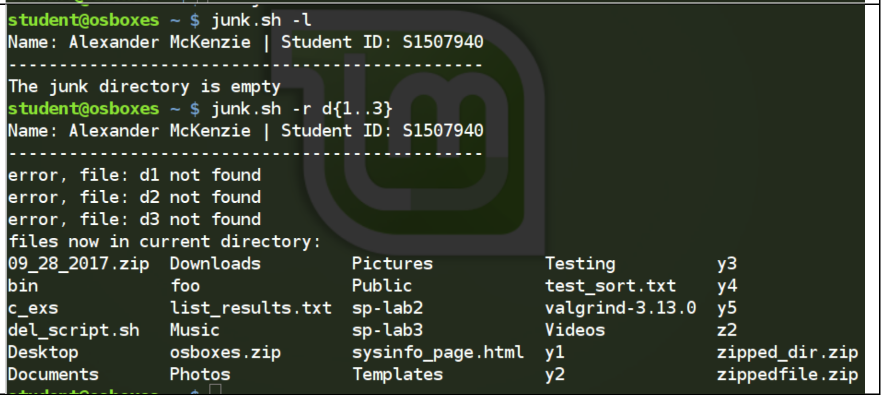


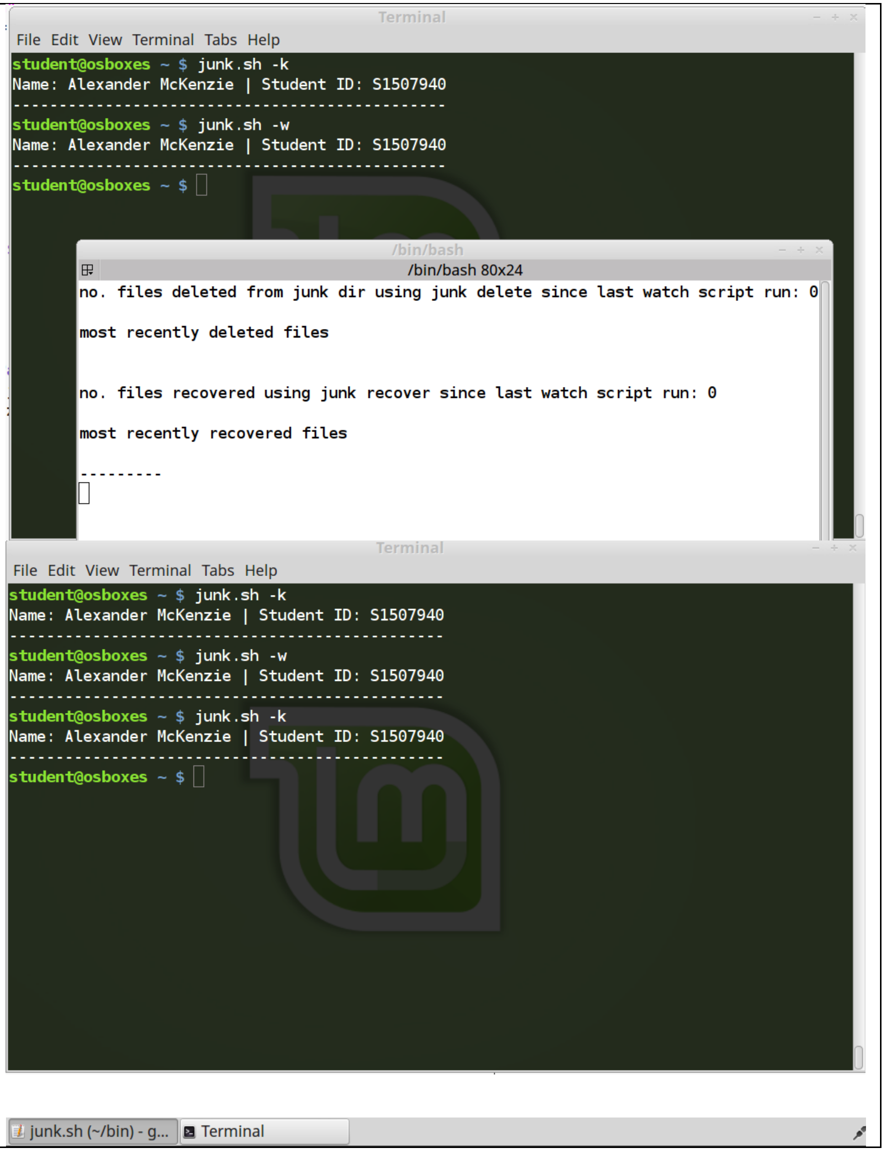
```
student@osboxes ~ $ ls .junkdir
student@osboxes ~ $ junk.sh -l
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
The junk directory is empty
student@osboxes ~ $
```



```
student@osboxes ~ $ junk.sh x{1..3}
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
x1 moved to junk directory
x2 moved to junk directory
x3 moved to junk directory
student@osboxes ~ $ junk.sh -l
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
filename: x1 | size in bytes: 0 | filetype: empty
filename: x2 | size in bytes: 0 | filetype: empty
filename: x3 | size in bytes: 0 | filetype: empty
```

</td>
</tr>
</table>

Recovering files from the junk directory	Normal	The junk script should recover files to the current directory	The junk script recovers specified files to the current directory	 student@osboxes ~ $ ls .junkdir y1 y2 y3 y4 y5 student@osboxes ~ $ junk.sh -r y{1..5} Name: Alexander McKenzie | Student ID: S1507940 -- y1 has been recovered to /home/student y2 has been recovered to /home/student y3 has been recovered to /home/student y4 has been recovered to /home/student y5 has been recovered to /home/student files now in current directory: 09_28_2017.zip Downloads Pictures Testing y3 bin foo Public test_sort.txt y4 c_exs list_results.txt sp-lab2 valgrind-3.13.0 y5 del_script.sh Music sp-lab3 Videos z2 Desktop osboxes.zip sysinfo_page.html y1 zipped_dir.zip Documents Photos Templates y2 zippedfile.zip student@osboxes ~ $
Recovering files that don't exist from the junk directory	Extreme	The junk script should output an error for each file not found in the junk directory	The junk script outputs an error for each file not found in the junk directory	 student@osboxes ~ $ junk.sh -l Name: Alexander McKenzie | Student ID: S1507940 -- The junk directory is empty student@osboxes ~ $ junk.sh -r d{1..3} Name: Alexander McKenzie | Student ID: S1507940 -- error, file: d1 not found error, file: d2 not found error, file: d3 not found files now in current directory: 09_28_2017.zip Downloads Pictures Testing y3 bin foo Public test_sort.txt y4 c_exs list_results.txt sp-lab2 valgrind-3.13.0 y5 del_script.sh Music sp-lab3 Videos z2 Desktop osboxes.zip sysinfo_page.html y1 zipped_dir.zip Documents Photos Templates y2 zippedfile.zip

Killing the watch script	Normal	The junk script should kill the current watch script process	The junk script was able to kill the current watch script process	

```
student@osboxes ~ $ junk.sh -k
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
student@osboxes ~ $ junk.sh -w
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
student@osboxes ~ $ 
```

```
no. files deleted from junk dir using junk delete since last watch script run: 0

most recently deleted files


no. files recovered using junk recover since last watch script run: 0

most recently recovered files

---------
```

```
student@osboxes ~ $ junk.sh -k
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
student@osboxes ~ $ junk.sh -w
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
student@osboxes ~ $ junk.sh -k
Name: Alexander McKenzie | Student ID: S1507940
-----------------------------------------------
student@osboxes ~ $ 
```