Report

Table of Contents

Rationale for My Own Commands

Why simplecalc is needed?

It makes it easier for users who are already using the terminal to calculate quick sums. It saves the user time because they no longer need to look around for a calculator. simplecalc reduces chance of an error occurring as the user of simplecalc no longer needs to work out the answer in their head.

How is simplecalc used?

When you enter the script you are required to enter the first number, then you choose the sign of the operation, and then you enter the last number. After carrying out these steps, an output will display the answer for the question. If you press the # it will terminate the script.

Why tempconv is needed?

It makes it more convenient for you, as you no longer need to look up a formula to convert between the two temperature units. This command can come in handy for scientists in particular, as it can convert their temperature readings for their science experiments.

How is tempconv used?

When you first enter the script you will be asked to choose what you would like to convert. You are then required to enter the temperature reading you would like to convert. Then, an output message appears showing you what the reading would be if you convert the temperature. If you press the # it would terminate the script.

Test Report

Introduction

I ensured my commands were working properly by carrying out different kinds of tests for each command, to identify if there were any errors. If there was an error, I removed it, so the code would run smoother. I didn't want an error to cause the program to crash.

To ensure that I had no errors I used white clear box testing, I incorporated conditions and control statements into my command.

tempconv

Negative test table:

Number	Test Case	Expected	Actual	Comments
1	-4.5C to F	23.9	Error	This command doesn't accept a floating point number; therefore, an error message appears. This is because I didn't include anything about floating point numbers my code.
2	4 to F		Help usage	If the user doesn't choose between C and F, I implemented a condition which would issue a help statement. This was because the code wouldn't know what it was converting if the user doesn't choose.
3	aC to F		Help usage	Inputting the letter "a", when choosing whether you want to convert Celsius to Fahrenheit or the other way around causes the help usage to come up. This is because "a" is not a type of temperature.
4	F to C	Error	-17	Even though the user didn't enter a number when asked for input the code assumes the input is 0. I should therefore add a help usage to say you have to enter a number.

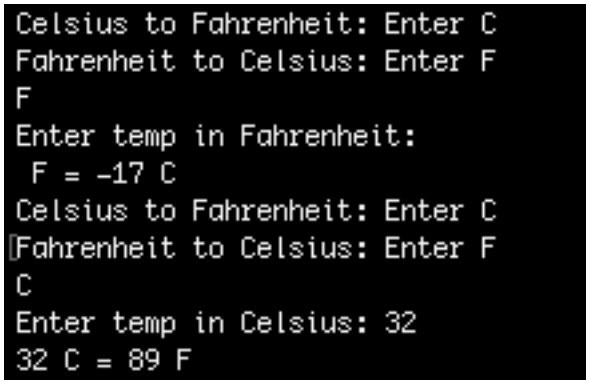
Positive test table:

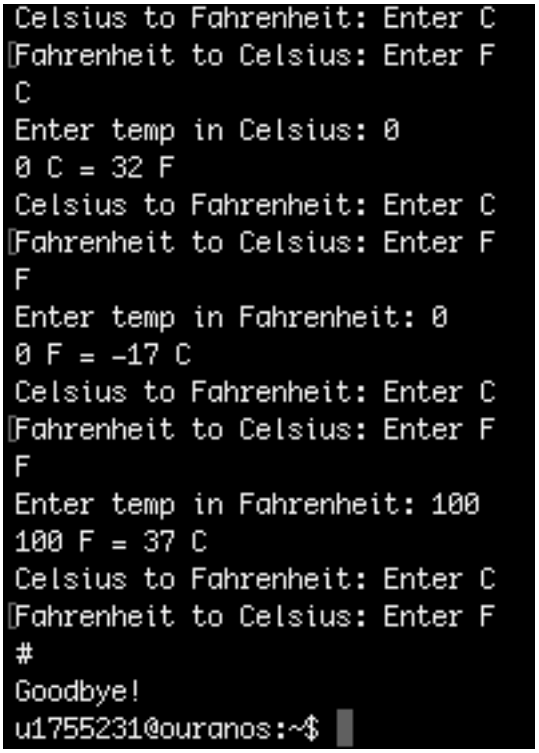
Number	Test Case	Expected	Actual	Comments
1	32C to F	89.6	89	The reason why I didn't get the exact value is because I haven't added anything about floating point numbers in the command.
2	0C to F	32	32	
3	0F to C	-17.7778	-17	
4	100F to C	37.7778	37	

Negative Test

```
[u1755231@ouranos:~$ tempconv                                                    ]
****************Welcome to the temperature converter.*********************
****************To exit the the converter press #.***********************
Celsius to Fahrenheit: Enter C
Fahrenheit to Celsius: Enter F
C
[Enter temp in Celsius: -4.5                                                     ]
/home/stud2/u1755231/bin/tempconv: line 20: let: -4.5: syntax error: invalid arithmetic operator (error
token is ".5")
-4.5 C =  F
Celsius to Fahrenheit: Enter C
Fahrenheit to Celsius: Enter F
```

Positive Test





---

simplecalc

Negative Test

Number	Test Case	Expected	Actual	Comments
1	4 * 4	16	Help usage: didn't enter the correct operator	A deficiency in my code was that if people preferred to use an (*) asterisk when multiplying numbers, they wouldn't be able to. But I have issued a help statement to tell the user to enter a X.
2	+		Error	Haven't added a conditional statement, for when the user doesn't enter a number. So I got a syntax error.
3	5 5		Help usage command: doesn't enter an operator.	You can't calculate the sum if you don't know the operator that is why a help usage is issued.

Positive Test

Number	Test Case	Expected	Actual
1	4 + 4	8	8
2	4 X 4	16	16
3	4 – 1	3	3
4	20 / 5	4	4

Negative Output

```
[u1755231@ouranos:~$ simplecalc
**********Welcome to the Simple Calculator**********
**********If you would like to leave the calculator type # **********
**********You can +, -, /, X between two numbers**********
[Number : 4
[Would you like to + -, /, X: *
[Number: 4
You didn't enter the right operator
NOTE: You need to enter a operator to calculate a sum
You also need to enter numbers when asked
If you would like to leave press #
[Number :
[Would you like to + -, /, X: +
[Number:
/home/stud2/u1755231/bin/simplecalc: line 43: let: answer= + : syntax error: operand expected (error tok
en is "+ ")
The answer is:
[Number : 5
[Would you like to + -, /, X:
[Number: 5
You didn't enter the right operator
NOTE: You need to enter a operator to calculate a sum
You also need to enter numbers when asked
If you would like to leave press #
```

Positive Output

```
[Number : 4
[Would you like to + -, /, X: +
[Number: 4
The answer is: 8
[Number : 4
[Would you like to + -, /, X: X
[Number: 4
The answer is: 16
[Number : 4
[Would you like to + -, /, X: -
[Number: 1
The answer is: 3
[Number : 20
[Would you like to + -, /, X: /
[Number: 5
The answer is: 4
Number :
```

Required Commands

masher.bash

I ensured that the users entered an argument, by adding a test to double check. I also added a check to make sure that the user inputs a string into the script, and I would issue a help statement if they didn't.

```
u1755231@ouranos:~/bin$ masher.bash
You should enter an argument!
Try entering an input now, if you would like to exit the string press
(full stop)
Enter string:
```

If the user enters my control character in the argument, then they would still enter the script even though according to assignment the script should terminate.

```
u1755231@ouranos:~/bin$ masher.bash hello/.
The value of masher is:  hello
Keep entering an input. If you would like to leave press a . (full stop)
When you leave it will display information on the string
Note: we do not count any special characters entered
Enter string:
```

Working command:

```
u1755231@ouranos:~/bin$ masher.bash fred\"123
The value of masher is:  fred123
Keep entering an input. If you would like to leave press a . (full stop)
When you leave it will display information on the string
Note: we do not count any special characters entered
Enter string: fred.fred
Control value is true at the postion: 5
Number of digits in the string: 0
Number of alphabetic characters in the string: 8
Number of non alpha numeric characters in the string: 1
Total number of characters in the string (including control): 9
Characters mashed before control value: 4
Total digits: 3
Total alphacharacters: 12
Total characters: 16
Total characters mashed before control value: 11
Number of loops: 1
u1755231@ouranos:~/bin$
```

chmx

If the file hasn't been executed a help statement would appear on the screen to explain why.

```
u1755231@ouranos:~/bin$ chmx hello
Sorry, can't find this file. Try again!
u1755231@ouranos:~/bin$ chmx
Need to enter a filename to make it executable to all.
u1755231@ouranos:~/bin$
```

Working command:

```
u1755231@ouranos:~/testfiles$ chmx fileout1
fileout1 has been made executable to all users
```

```
-rw-r--r-- 1 u1755231 stud1617    5 Oct 18 08:31 fileout1
```

```
u1755231@ouranos:~/testfiles$ ls -l
total 52
-rw-r--r-- 1 u1755231 stud1617    0 Nov 23 14:20 file
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file1
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file2
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file3
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file4
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file5
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file6
-rwxr-xr-x 1 u1755231 stud1617   15 Oct 18 08:31 file7
-rwxr-xr-x 1 u1755231 stud1617    5 Oct 18 08:31 fileout1
```

lshead

If the user entered no argument an error would appear; this was because I had not added a check for this. Another problem was that lshead only found directories that were in the home directory.

```
u1755231@ouranos:~/studio1$ lshead
/home/stud2/u1755231/bin/lshead: line 11: [: ==: unary operator expected
/home/stud2/u1755231/bin/lshead: line 19: [: ==: unary operator expected
Can't find this directory in your home page
You also need to enter head or tail. Make sure you put a - before
like this -head
```

```
u1755231@ouranos:~/studio1$ lshead -head 3 file
Can't find this directory in your home page
You also need to enter head or tail. Make sure you put a - before
like this -head
u1755231@ouranos:~/studio1$
```

Working Command:

```
u1755231@ouranos:~$ lshead -head 3 bin
==> chmx <==
#! /bin/bash
#chmx -  a command that takes filenames as its arguments and makes those fi$
# executable to all user

==> del <==
#! /bin/bash
# del - a version of rm which only copies files to a special ".waste"
# directory.

==> emptywaste <==
#! /bin/bash
#emptywaste: A command which deletes everything for good in the waste
#bin

==> listwaste <==
#! /bin/bash
#listwaste: a command which lists the name of all the files in your waste
#bin and their size in bytes

==> lshead <==
#! /bin/bash
#ishead is a command that lists the first few lines of every files in a
#directory specified by the argument. This command can also list the first n

==> masher.bash <==
#! /bin/bash

echo "The command masher.bash allows you to input any characters you like."

==> simplecalc <==
#! /bin/bash
#simplecalc: is a command which can +, -, /, X between two numbers
#author Amrit Sarai Nov 2017

==> tempconv <==
#! /bin/bash
# converter: Is a command which can convert celsius to fahrenheit
# author Amrit Sarai Nov 2017

==> wastesize <==
#! /bin/bash
# wastesize: A command which reports the number of files in the waste bin.
# author Amrit Sarai Nov 2017
```

```
u1755231@ouranos:~$ cd bin
u1755231@ouranos:~/bin$ ls
chmx  del  emptywaste  listwaste  lshead  masher.bash  simplecalc  tempconv  wastesize
u1755231@ouranos:~/bin$
```

```
  GNU nano 2.2.6                                              File: chmx

#! /bin/bash
#chmx -  a command that takes filenames as its arguments and makes those fi$
# executable to all user
# author Amrit Sarai Nov 2017
```

```
Last login: Thu Dec 14 00:15:23 2017 from selene
[u1755231@ouranos:~$ lshead -tail 3 bin
==> chmx <==
     echo "Sorry, can't find this file. Try again!"
fi
done

==> del <==
     echo "the waste bin."
     fi
done

==> emptywaste <==

rm -rf /home/stud2/u1755231/.waste/*
echo "Everything has been removed from the waste bin"

==> listwaste <==

ls -l /home/stud2/u1755231/.waste


==> lshead <==
    echo "like this -head"
fi


==> masher.bash <==
    fi
done


==> simplecalc <==
         echo "If you would like to leave press #"
     fi
done

==> tempconv <==
       echo "To exit the converter press #"
   fi
done

==> wastesize <==
#excluding its own directory
find /home/stud2/u1755231/.waste/* | wc -l

u1755231@ouranos:~$
```

del

The command checks to see if the file exists, and is readble before trying to copy it to the waste bin.

```
u1755231@ouranos:~/studio1$ del blah
The file blah does not exist or it is not readable.
Please note that directories can't be added to (.waste)
the waste bin.
```

Working command:

```
u1755231@ouranos:~/studio1$ ls
dfile  myfile.txt  testfiles  whateveryouwanttocallit
u1755231@ouranos:~/studio1$ del dfile
file has been copied into waste bin
u1755231@ouranos:~/studio1$ del myfile.txt
file has been copied into waste bin
u1755231@ouranos:~/studio1$
```

```
u1755231@ouranos:~/.waste$ ls -l
total 4
-rwxr--r-- 1 u1755231 stud1617 24 Dec 13 00:44 dfile
-rwxr-xr-x 1 u1755231 stud1617  0 Dec 13 00:45 myfile.txt
u1755231@ouranos:~/.waste$
```

---

wastesize

When there was nothing in my waste bin I would get an error. This was because if I wanted to count the number of files, and not the directory itself, I had to add a * at the end of my path. So when everything was removed from the waste bin, the command couldn't find the path for the waste bin.

```
u1755231@ouranos:~$ cd .waste
u1755231@ouranos:~/.waste$ ls
u1755231@ouranos:~/.waste$
```

```
u1755231@ouranos:~$ wastesize
find: '/home/stud2/u1755231/.waste/*': No such file or directory
0
u1755231@ouranos:~$
```

Working command :

```
u1755231@ouranos:~/.waste$ ls -l
total 4
-rwxr--r-- 1 u1755231 stud1617 24 Dec 13 00:44 dfile
-rwxr-xr-x 1 u1755231 stud1617  0 Dec 13 00:45 myfile.txt
u1755231@ouranos:~/.waste$
```

```
u1755231@ouranos:~/studio1$ wastesize
2
u1755231@ouranos:~/studio1$
```

---

emptywaste

I didn't have a check in my command which would check whether the bin was empty before trying to remove everything from it. Therefore, if there was nothing to be removed, I would still get a message saying everything has been removed.

```
u1755231@ouranos:~$ cd .waste
u1755231@ouranos:~/.waste$ ls
u1755231@ouranos:~/.waste$
```

```
u1755231@ouranos:~$ emptywaste
Everything has been removed from the waste bin
u1755231@ouranos:~$
```

Working command:

```
-rwxr-xr-x 1 u1755231 stud1617  0 Dec 13
u1755231@ouranos:~/.waste$ ls
dfile  myfile.txt
u1755231@ouranos:~/.waste$
```

```
u1755231@ouranos:~/.waste$ emptywaste
Everything has been removed from the waste bin
u1755231@ouranos:~/.waste$
```

---

listwaste

I didn't add any tests for this command because If the bin was empty then a message would appear anaywas saying there is 0 files.

```
u1755231@ouranos:~$ cd .waste
u1755231@ouranos:~/.waste$ ls
u1755231@ouranos:~/.waste$
```

```
Everything has been removed fro
u1755231@ouranos:~$ listwaste
total 0
u1755231@ouranos:~$
```

Working command: