

Will Townsend

Task 1:

I tested every variable as seen below and the ones that are empty make logical sense.

```
File Edit View Bookmarks Settings Help
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w\$
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $EDITOR

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $HOME
/mnt/linuxlab/home/wtownsend2
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $HOSTNAME
hpc13-5
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $LD_LIBRARY_PATH

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $LESS

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $MAIL

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $MANPATH
:/opt/puppetlabs/puppet/share/man
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $MORE

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $PAGER

wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
:/bin:/opt/puppetlabs/bin
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $PWD
/mnt/linuxlab/home/wtownsend2/COSC350/Lab03
wtownsend2@hpc13-5:~/COSC350/Lab03$ $SHELL
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $SHELL
/bin/bash
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $TERM
xterm-256color
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo $USER
wtownsend2
wtownsend2@hpc13-5:~/COSC350/Lab03$
```

Task 2:

```
File Edit View Bookmarks Settings Help
wtownsend2@hpc13-5:~/COSC350/Lab03$ temp=$PS1
wtownsend2@hpc13-5:~/COSC350/Lab03$ PS1="\[[COSC350 \W]\] "
[COSC350 Lab03] PS1=$temp
wtownsend2@hpc13-5:~/COSC350/Lab03$
```

Task 3:

```
File Edit View Bookmarks Settings Help
wtownsend2@hpc13-5:~/COSC350/Lab03$ cd ../../
wtownsend2@hpc13-5:~$ ls -l D
Desktop/ Documents/ Downloads/
wtownsend2@hpc13-5:~$ cd COSC350
wtownsend2@hpc13-5:~/COSC350$ ls -l Lab
Lab01/ Lab02/ Lab03/ Lab04/ Lab05/ Lab06/ Lab07/ Lab08/ Lab09/ Lab10/ Lab11/ Lab12/
wtownsend2@hpc13-5:~/COSC350$ ls -l Lab
Lab01/ Lab02/ Lab03/ Lab04/ Lab05/ Lab06/ Lab07/ Lab08/ Lab09/ Lab10/ Lab11/ Lab12/
wtownsend2@hpc13-5:~/COSC350$ cd ..
wtownsend2@hpc13-5:~$ cd COSC220
wtownsend2@hpc13-5:~/COSC220$ ls -l Lab
Lab1/ Lab2/ Lab3/ Lab4/
wtownsend2@hpc13-5:~/COSC220$ cd ..
wtownsend2@hpc13-5:~$
```

Task 4:

As seen in the screenshot when the contents of foo are sorted it sorts by ascending numerical order.

```
File Edit View Bookmarks Settings Help
wtownsend2@hpc13-5:~/COSC350/Lab03$ ls Test
ls: cannot access 'Test': No such file or directory
wtownsend2@hpc13-5:~/COSC350/Lab03$ ls Test 2> bar
wtownsend2@hpc13-5:~/COSC350/Lab03$ ls Test 2> /dev/null
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo '3'>foo
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo '5'>>foo
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo '2'>>foo
wtownsend2@hpc13-5:~/COSC350/Lab03$ echo '1'>>foo
wtownsend2@hpc13-5:~/COSC350/Lab03$ cat foo>bar
wtownsend2@hpc13-5:~/COSC350/Lab03$ cat foo<sort
bash: sort: No such file or directory
wtownsend2@hpc13-5:~/COSC350/Lab03$ sort<foo
1
2
3
5
wtownsend2@hpc13-5:~/COSC350/Lab03$ sort<foo>bar
wtownsend2@hpc13-5:~/COSC350/Lab03$
```

Task 5:

we outputs the following information: print newline count, word count and byte count so 100 new lines 100 'words' and 292 byte count in numbs and 14 newline count, 14 word count, and 42 byte count

```
File Edit View Bookmarks Settings Help
wtownsend2@hpc13-5:~/COSC350/Lab03$ for i in $(seq 1 100);do echo "$i">>numbs; done
wtownsend2@hpc13-5:~/COSC350/Lab03$ wc numbs
100 100 292 numbs
wtownsend2@hpc13-5:~/COSC350/Lab03$ sed -n '25,38p;39q' numbs>somenums
wtownsend2@hpc13-5:~/COSC350/Lab03$ wc somenums
14 14 42 somenums
wtownsend2@hpc13-5:~/COSC350/Lab03$
```

Task 6a:

```
#!/bin/bash
```

```
base=0
```

```
num=0
```

```
while [ $base -lt 4 ] || [ $num -eq 0 ];do
```

```
    echo -n "Base Value: "
```

```
    read base
```

```
    num=`expr $base % 2`
```

```
    if [ $base -lt 4 ];then
```

```
        echo "Base Value too small!"
```

```
    elif [ $num -eq 0 ];then
```

```
        echo "Base Value must be odd!"
```

```
    fi
```

```
done
```

```
half=`expr $base / 2`
```

```
for i in `seq $half -1 1`;do
```

```
    for j in `seq 1 $i`;do
```

```
        echo -n ' '
```

```
    done
```

```
    star=`expr $half - $i`
```

```
    star=`expr 2 \* $star + 1`
```

```
    for j in `seq 1 $star`;do
```

```
        echo -n '*'
```

```
    done
```

```
    echo ' '
```

```
done
```

```
for i in `seq 1 $base`;do
```

```
    echo -n '*'
```

```
done
```

```
echo
```

```
for i in `seq 1 $half`;do
```

```
    for j in `seq 1 $i`;do
```

```
        echo -n ' '
```

```
    done
```

```
    star=`expr $half - $i`
```

```
    star=`expr 2 \* $star + 1`
```

```
    for j in `seq 1 $star`;do
```

```
        echo -n '*'
```

```
    done
```

```
    echo ' '
```

```
done
```

```
echo
```

```
exit 0
```

Task 6b:

```
#!/bin/bash
```

```
base=0
```

```
num=0
```

```
tail=0
```

```
while [ $base -lt 4 ] || [ $num -eq 0 ];do
    echo -n "Base Value: "
    read base
    num=`expr $base % 2`
    if [ $base -lt 4 ];then
        echo "Base Value too small!"
    elif [ $num -eq 0 ];then
        echo "Base Value must be odd!"
    fi
done
```

```
while [ $tail -le 0 ];do
    echo -n "Tail value: "
    read tail
    if [ $tail -le 0 ];then
        echo "Tail value must be positive!"
    fi
done
```

```
tail=`expr $base + $tail`
half=`expr $base / 2`
for i in `seq 1 $half`;do
    space=`expr $base - 2 \* $i`
    for j in `seq 1 $space`;do
        echo -n ' '
    done
    star=`expr 2 \* $i - 1`
    for j in `seq 1 $star`;do
        echo -n '*'
    done
    echo
done
for i in `seq 1 $tail`;do
    echo -n '*'
done
echo
for i in `seq $half -1 1`;do
    space=`expr $base - 2 \* $i`
    for j in `seq 1 $space`;do
        echo -n ' '
    done
    star=`expr 2 \* $i - 1`
    for j in `seq 1 $star`;do
        echo -n '*'
    done
done
```

```
    echo  
done  
exit 0
```

Task 6c:

```
#!/bin/bash
```

```
base=0
```

```
num=0
```

```
tail=0
```

```
while [ $base -lt 4 ] || [ $num -eq 0 ];do
    echo -n "Base Value: "
    read base
    num=`expr $base % 2`
    if [ $base -lt 4 ];then
        echo "Base Value too small!"
    elif [ $num -eq 0 ];then
        echo "Base Value must be odd!"
    fi
done
```

```
while [ $tail -le 0 ];do
    echo -n "Tail value: "
    read tail
    if [ $tail -le 0 ];then
        echo "Tail value must be positive!"
    fi
done
```

```
half=`expr $base / 2 - 1`
for i in `seq 0 $half`;do
    for j in `seq 1 $tail`;do
        echo -n ' '
    done
    star=`expr 2 \* $i + 1`
    for j in `seq 1 $star`;do
        echo -n '*'
    done
    echo
done
mid=`expr $tail + $base`
for i in `seq 1 $mid`;do
    echo -n '*'
done
echo
for i in `seq $half -1 0`;do
    for j in `seq 1 $tail`;do
        echo -n ' '
    done
    star=`expr 2 \* $i + 1`
    for j in `seq 1 $star`;do
        echo -n '*'
    done
    echo
done
```

exit 0

Task 7:

```
#!/bin/bash
```

```
echo "Task 7:"
```

```
echo -n "Enter a value: "
```

```
read num
```

```
echo -n "$num!="
```

```
res=1
```

```
while [ $num -gt 1 ]
```

```
do
```

```
    res=`expr $res \* $num`
```

```
    let num--
```

```
done
```

```
echo "$res"
```

```
exit 0
```


Task 8:

```
#!/bin/bash
```

```
echo "Task 8:"
```

```
res=0
```

```
if [ "$#" -eq 1 ];then
```

```
    len=`expr length $1`
```

```
    for i in `seq 1 $len`;do
```

```
        digit=`expr substr $1 $i 1`
```

```
        res=`expr $res + $digit`
```

```
    done
```

```
    echo "The sum of each digit in $1 is $res"
```

```
else
```

```
    echo "Must pass one parameter!"
```

```
fi
```

```
exit 0
```

Task 9:

```
#!/bin/bash
```

```
echo "Task 9:"
```

```
echo -n "Please enter the directory path (path/to/file/from/root no ~): "
```

```
read dir
```

```
path=$PWD
```

```
cd /
```

```
if [ -d $dir ];then
```

```
    cd $dir
```

```
    num=3
```

```
    while [ $num -gt 0 ];do
```

```
        echo -n "Please enter the file name: "
```

```
        read file
```

```
        if [ -f $file ] && [ -r $file ];then
```

```
            echo -n "Please enter a word to search for: "
```

```
            read word
```

```
            if grep -q $word $file
```

```
            then
```

```
                echo "$word was found!"
```

```
                cd /
```

```
                cd $path
```

```
                exit 0
```

```
            else
```

```
                echo "$word not found!"
```

```
                cd /
```

```
                cd $path
```

```
                exit 4
```

```
            fi
```

```
        else
```

```
            let num--
```

```
            echo "File does not exist or is not readable ($num more attempts)."
```

```
        fi
```

```
        if [ ! -f $file ] && [ $num -eq 0 ];then
```

```
            cd $path
```

```
            exit 2
```

```
        elif [ -r $file ] && [ $num -eq 0 ];then
```

```
            cd $path
```

```
            exit 3
```

```
        fi
```

```
    done
```

```
    exit 2
```

```
else
```

```
    echo "This is not an existing directory."
```

```
    cd $path
```

```
    exit 1
```

```
fi
```