COSC 350 System Software: Mini Test #1

09/18/20

Name: Jacob Duncan

1. (1 pt.) Create a file named **numbs** that contains the integers 1 through 100, one integer per line with shell commends with output redirection. The file will have 100 lines. You need use for loop.

   For I in {1..100}; do
   Echo $I >> numbs
   Done

```
❯ for i in {1..100}; do
for> echo $i >> numbs
for> done
```

2. (1 pt.) There are two types of libraries: static library and shared library. Briefly explain differences between static and shared library.

   - **Static library: (.s) Also called archives are a collection of object files in a ready to use form. To use these libraries, you must include a header file in your program.**

   - **Shared library: (.ao) The libraries that can be linked to any program at run-time. Once they are loaded once, they can be used on any program because there is always a single copy of a shared library in memory. These types of libraries save space.**

3. (1 pt.) When we pass arguments to a script, system save each arguments in positional parameters $0, $1, $2, … What information will save on each of following positional parameters

   - **$@ : List all the parameters in a single variable**
   - **$*: List all the parameters in a single variable**
   - **$0: First parameter passed (usually the name of the executable)**
   - **$#: Number of parameters passed**

4. (1 pt.) What are two conditions to make a shell script file executable

1. **Provide where the bash shell program is located (!# /bin/bash)**

2. **Change the script mode to executable (chmod +x ./file_name.sh)**

5. (2 pt.) Since a directory itself is a file in Linux system, each directory has name. Write bash script which test each file names in current directory and display subdirectory names.

```
#!/bin/bash

For file in *; do
    If [ -d “$file” ]; then
        Echo “$file”
    Fi
Done
Exit 0
```

I coded it in my terminal and took a picture and attached it because the capitalization is strange in Microsoft Word.

```
#!/bin/bash

for file in *; do
  if [ -d "$file" ]; then
    echo "$file"
  fi
done
exit 0
```

6. (2 pt.) Write a script to calculating factorial of given number by using while loop. The script asks an integer value with read command and calculates factorial and display the result.

Prompt=”Enter a number to calculate the factorial of (must be positive): “
Echo –n “$prompt”
Read input

Factorial = 1
While [“$input” -lt 0]; do

Echo "invalid entry, you must enter a positive number"
Echo –n "$prompt"
Read input
Done

Temp=$input

While [$input –gt 0]; do
Factorial = $(($factorial * $input))
Input=$(($input –1))
Done

Echo "The factorial of $temp is $factorial"
Exit 0

I coded it in my terminal and took a picture and attached it because the capitalization is strange in Microsoft Word.

```
prompt="Enter a number to calculate the factorial of (must be positive): "
echo -n "$prompt"
read input

factorial=1

while [ "$input" -lt 0 ]; do
  echo "Invalid entry, you must enter a positive integer"
  echo -n "$prompt"
  read input
done

temp=$input

while [ $input -gt 0 ]; do
  factorial=$(($factorial * $input))
  input=$(($input -1))
done

echo "$temp! = $factorial"

exit 0
```

7. (1 pt.) Write shell script by using nested for loop to print the following patterns on screen based on an integer input n (between 1 and 9) from the keyboard. (Do not use (()) in for loop). Your program display following shape with input 6.

1
22
333
4444
55555

666666

Echo –n "Input number between 1 and 9: "
Read num

While [$num –lt 1 –o $num –gt 9]; do
Echo –n "Error, enter a number between 1 and 9: "
Read num
Done

For I in $(seq 1 $num); do
For j in $(seq 1 $I); do
Echo –n "$I"
Done
Echo
Done
Exit 0

I coded it in my terminal and took a picture and attached it because the capitalization is strange in Microsoft Word.

```
echo -n "Input number between 1 and 9: "
read num

while [ $num -lt 1 -o $num -gt 9 ]; do
  echo -n "Error, enter a number between 1 and 9: "
  read num
done

for i in $(seq 1 $num); do
  for j in $(seq 1 $i); do
    echo -n "$i"
  done
  echo
done
exit 0
```

8. (1 pt.) Briefly explain the difference between the following two bash commands:

ls -l | less : **This command will execute the ls –l command and display it in page format where we can view the files and directories page by page.**

ls -l > less : **This command will execute the ls –l command and redirect the output to a file named less.**