Imani Martinez

Lab Assignment 4

1. Write a bash script that prints the following five string variables (1 pt) and the length of all of the variables added together (1 pt).

```
#!/bin/bash

# Assigning variables
string_v="Methionine"
string_v2="Leucine"
string_v3="Cysteine"
string_v4="Alanine"
string_v5="Valine"

# Printing variables
echo "$string_v"
echo "$string_v2"
echo "$string_v3"
echo "$string_v4"
echo "$string_v5"
echo    # Print an empty line

# Length of all variables added together
total_length=$((${#string_v}+${#string_v2}+${#string_v3}+${#string_v4}+${#string_v5}))
echo "The total length of all the variables added together is $total_length"
echo    # Print an empty line
```

Imani Martinez

2. Write a bash script to count the number of start codons (ATG) (1 pt) and stop codons (TAA, TAG, TGA) (1 pt) from the example2.fasta file.

- **Please use variables for the start and stop codons and print out the count in a meaningful way.**
- **Remember that start codons only occur at the beginning of sequences and stop codons only occur at the end of sequences.**

```
#!/bin/bash

# Define the input FASTA file
input_file="example2.fasta"

# Assigning variables
start_codon="ATG"
stop_codons="TAA|TAG|TGA"

# Count the number of start codons (ATG)
start_count=$(grep -o -E "^$start_codon" $input_file | wc -l)

# Count the number of stop codons (TAA, TAG, TGA)
stop_count=$(grep -o -E "$stop_codons$" $input_file | wc -l)

# Print the results
echo "Number of start codons ($start_codon) is $start_count"
echo "Number of stop codons ($stop_codons) is $stop_count"
```

3. Write a bash script that prints the following:

- **Username (0.25 pt)**
- **Current directory (0.25 pt)**
- **Location of root directory (0.25 pt)**
- **Date/time (0.25 pt)**

```
 #!/bin/bash


whoami
pwd
echo $ROOT
date
```

Imani Martinez

4. Do the following commands compress or uncompress a file?

- **gunzip file.gz (0.25 pt)** : uncompress a file
- **tar -zxvf file.tar.gz (0.25 pt)** : uncompress a file
- **zip file.zip file.txt file1.txt (0.25 pt)** : compress a file
- **tar -zcvf file.tar.gz file.txt file1.txt (0.25 pt)** : compress a file

5. Write an array in bash that contains these amino acids (0.25 pt):

- **Methionine**
- **Leucine**
- **Cysteine**
- **Alanine**
- **Valine**
- **Tyrosine**
- **Proline**

```
#!/bin/bash

array=( "Methionine" "Leucine" "Cysteine" "Alanine" "Valine" "Tyrosine" "Proline")
```

Give the command to delete Alanine (0.25 pt)

```
# Print out array with "Alanine" deleted

echo "Delete Alanine"
echo ${array[@]/"Alanine"}
echo    # Print an empty line
```

Give the command to have print the aminos from Cysteine to Tyrosine (0.25 pt)

```
echo "The third through sixth elements are: ${array[@]:2:5}"
```

Give the command to add Histidine to the array (0.25 pt)

```
# Add "Histidine" top array

echo "Add Histidine"
array=("${array[@]}" "Histidine")
echo ${array[@]}
```

6. Give the command to count how many times the name 'abdul' is left to the name 'chi' in doppelganger_names.txt (0.5 pt). What is the count (0.5 pt)?

```
awk '/abdul\tchi/' doppelganger_names.txt | wc -l
```

The count is 8.

7. Give an example of a legal variable name (0.5 pt) and an illegal variable name (0.5 pt).

Legal Variable Name: myvar

Illegal Variable Name: 1myvar

8. What are the commands to "compile" (0.5 pt) and run a bash script (0.5 pt)?

Compile with chmod: chmod a+x file.sh

Run with: ./file.sh Or bash file.sh

Bonus I (2 pts): Write a bash script to iterate through the amino acid array.

- **Methionine**
- **Leucine**
- **Cysteine**
- **Alanine**
- **Valine**
- **Tyrosine**
- **Proline**

```
#!/bin/bash


# Define the amino acid array
amino_acids=("Methionine" "Leucine" "Cysteine" "Alanine" "Valine" "Tyrosine" "Proline")


# Iterate through the array and print each amino acid
for amino_acid in "${amino_acids[@]}"; do
    echo "$amino_acid"
```

Imani Martinez

Bonus II (1 pt): Do Bonus I in any other language.

Python:

```
# Define the amino acid array
amino_acids = ["Methionine", "Leucine", "Cysteine", "Alanine", "Valine", "Tyrosine", "Proline"]

# Iterate through the array and print each amino acid
for amino_acid in amino_acids:
    print(amino_acid)
```

Bonus III (2 pts): Convert all of example2.fasta into its amino acids in a bash script. You may use any commands you want. If you did the similar bonus (Bonus II) last week, you are NOT allowed to do it again.