IDC306 - Assignment 1

Rishi Vora MS21113

Q1. Shell program to check for name and password match input by a user (the input text for password will not be visible).

```shell
#!/bin/bash

username="admin"
password="somepassword"

read -p "Enter username: " input_username
read -sp "Enter password: " input_password

if [ $input_username === $username ] && [ $input_password === $password ]; then
    echo -e "\n\nUsername and password match"
else
    echo -e "\n\nUsername and password do not match!"
fi
```

Q2. User can input password only a fixed number of times (3) or until it gets correct.

```shell
#!/bin/bash

username="admin"
password="somepassword"

read -p "Enter username: " input_username

if [ $input_username != $username ]; then
    echo -e "\n\nUsername is incorrect. Exiting..."
    exit 1
fi

read -sp "Enter password: " input_password

tries=1
max_tries=3

while [[ $input_password != $password ]]; do
    if [ $tries -lt $max_tries ]; then
        echo -e "\n\nPassword is incorrect. You have $((max_tries - tries)) tries left."
        ((tries++))

        read -sp "Enter password: " input_password
```

```
    else
        echo -e "\n\nYou have exceeded the maximum number of tries. Exiting..."
        exit 1
    fi
done
```

Q3. Practice loops for performing addition of even/odd numbers.

Shell

```
#!/bin/bash

sum_even=0

for i in {1..10}; do
    if (( i%2 === 0 )); then
        sum_even=$((sum_even + i))
    fi
done

echo "Sum of even numbers from 1 to 10: $sum_even"


sum_odd=0

for i in {1..10}; do
    if (( i%2 != 0 )); then
        sum_odd=$((sum_odd + i))
    fi
done

echo "Sum of odd numbers from 1 to 10: $sum_odd"
```

Q4. Find the number of sequences in the fasta file. Compute the composition of A/T/G/C for each sequence.

Shell

```
#!/bin/bash

file=./fasta_file.txt

if ! [ -f $file ]; then
    echo "$file does not exist. Exiting..."
    exit 1
fi

declare -A sequences

while read line; do
```

```
    if [[ ${line:0:1} === ">" ]]; then
        seqid=${line:1:14}
    else
        sequences[$seqid]=${sequences[$seqid]}$line
    fi
done < $file

echo -e "There are ${#sequences[@]} sequences in $file\n"

for seqid in ${!sequences[@]}; do
    seq=${sequences[$seqid]}
    seqlen=${#seq}
    declare -A counts=([A]=0 [T]=0 [G]=0 [C]=0)

    for i in `seq 0 $((seqlen-1))`; do
        ch=${seq:$i:1}
        ((counts[$ch]+=1))
    done

    echo -e "Composition of $seqid:\nA: ${counts[A]}, T: ${counts[T]}, G:
    ${counts[G]}, C: ${counts[C]}"
done
```

Q5. Write a calculator function to perform addition, and subtraction based on user inputs.

Shell

```
#!/bin/bash

read -p "Enter operation (add or sub): " operation
if [[ $operation != "add" && $operation != "sub" ]]; then
    echo "Invalid operation."
    exit 1
fi

read -p "Enter first number: " num1
read -p "Enter second number: " num2

case "$operation" in
    "add")
        echo "Result: $((num1 + num2))"
    ;;
    "sub")
        echo "Result: $((num1 - num2))"
    ;;
esac
```

Q6. Write a program to find an input string that is palindrome.

Shell

```
#!/bin/bash

read -p "Give input string: " input

len=${#input}
revd=""

for (( i=$len-1; i>=0; i-- )); do
    revd=$revd${input:$i:1}
done

if [[ $input === $revd ]]; then
    echo -e "\nIt is a palindrome!"
else
    echo -e "\nNot a palindrome"
fi
```

Q7. Write the complement of the sequence in the fasta file.

Shell

```
#!/bin/bash

file=./fasta_file.txt

if ! [ -f $file ]; then
    echo "$file does not exist. Exiting..."
    exit 1
fi

declare -A sequences

while read line; do
    if [[ ${line:0:1} === ">" ]]; then
        seqid=${line:1:14}
    else
        sequences[$seqid]=${sequences[$seqid]}$line
    fi
done < $file

declare -A complements=([A]=T [T]=A [G]=C [C]=G)

for seqid in ${!sequences[@]}; do
    seq=${sequences[$seqid]}
    seqlen=${#seq}
    comp=""

```

```
    for i in `seq 0 $seqlen`; do
        ch=${seq:$i:1}

        comp=$comp${complements[$ch]}
    done

    echo -e "Complementary sequence of $seqid is\n$comp"
done
```

Q8. Find the the ORF in the input sequence of the fasta file. The ORF starts with 'ATG/ GTG' and terminates with TAG/TAA/TGA.

Shell

```
#!/bin/bash

file=./fasta_file.txt

if ! [ -f $file ]; then
    echo "$file does not exist. Exiting..."
    exit 1
fi

declare -A sequences

while read line; do
    if [[ ${line:0:1} === ">" ]]; then
        seqid=${line:1:14}
    else
        sequences[$seqid]=${sequences[$seqid]}$line
    fi
done < $file

start_codons=("ATG" "GTG")
stop_codons=("TAG" "TAA" "TGA")

for seqid in ${!sequences[@]}; do
    seq=${sequences[$seqid]}
    seqlen=${#seq}

    for frame in 0 1 2; do
        i=$frame

        while [[ $i -lt $((seqlen-2)) ]]; do
            start_codon="${seq:$i:3}"

            if [[ ${start_codons[@]} =~ $start_codon ]]; then

```

```
                for (( j=$i; j<=$seqlen-2; j+=3)); do
                    stop_codon="${seq:$j:3}"

                    if [[ ${stop_codons[@]} =~ $stop_codon ]]; then
                        orf="${seq:$i:$((j - i + 3))}"
                        echo -e "\nORF found in $seqid in frame $((frame+1))
                        from $((i+1)) to $((j+3)):\n$orf"
                        i=$j
                        break
                    fi
                done
            fi
            i=$((i+3))
        done
    done
done
```