Operating Systems Laboratory (CSE 4510) - I

Bash Practice Problems: Conditionals, Loops & Functions

Conditional Statements

Problem 1: Write a script that asks the user to enter a number and checks if it's positive, negative, or zero.
Solution:

```
#!/bin/bash
read -p "Enter a number: " num
if (( num > 0 )); then
  echo "Positive"
elif (( num < 0 )); then
  echo "Negative"
else
  echo "Zero"
fi
```

Problem 2: Write a script that takes a filename as input and checks:

- If it exists
- If it's readable
- If it's a regular file and not a directory

Solution:

```
#!/bin/bash
read -p "Enter filename: " fname
if [[ -f "$fname" && -r "$fname" ]]; then
  echo "File exists and is readable"
elif [[ -d "$fname" ]]; then
  echo "It's a directory"
else
  echo "Not found or not accessible"
fi
```

Problem 3: Write a script to check if a given year is a leap year.

Solution:

```
#!/bin/bash
read -p "Enter a year: " year
if (( year % 400 == 0 )); then
    echo "$year is a leap year"
elif (( year % 100 == 0 )); then
    echo "$year is not a leap year"
elif (( year % 4 == 0 )); then
    echo "$year is a leap year"
else
    echo "$year is not a leap year"
fi
```

Loops

Problem 1: Write a script that prints numbers from 1 to 10 using a for loop.

Solution:

```
#!/bin/bash
for i in {1..10}; do
  echo "Number: $i"
done
```

Problem 2: Write a script that reads a file line by line and prints each line with a line number.

Solution:

```
#!/bin/bash
counter=1
while read line; do
  echo "$counter: $line"
  ((counter++))
done < "$1"
```

Problem 3: Write a script that keeps asking for a password until the user enters "secret123". Use an until loop.

Solution:

```
#!/bin/bash
pass=""
until [[ "$pass" == "secret123" ]]; do
  read -sp "Enter password: " pass
  echo
done
echo "Access granted!"
```

Problem 4: Write a script to print all even numbers from 1 to a given number N.

Solution:

```
#!/bin/bash
read -p "Enter a number: " N
for (( i=1; i<=N; i++ ))
do
    if (( i % 2 == 0 )); then
        echo $i
    fi
done
```

Problem 4: Write a script that prints all prime numbers between 1 and a given number N.

Solution:

```
#!/bin/bash
read -p "Enter a number: " N
for (( num=2; num<=N; num++ ))
do
    is_prime=1
    for (( i=2; i*i<=num; i++ ))
    do
        if (( num % i == 0 )); then
            is_prime=0
            break
        fi
    done
    if (( is_prime == 1 )); then
        echo $num
    fi
done
```

Functions

Problem 1: Write a function called greet that takes a name and prints:"Hello, [name]! Welcome to Bash scripting."

Solution:

```
#!/bin/bash
greet() {
  echo "Hello, $1! Welcome to Bash scripting."
}
greet "Enamul"
```

Problem 2: Write a function is_even that takes a number and returns 0 if even, 1 if odd.

Solution:

```
#!/bin/bash
is_even() {
  if (( $1 % 2 == 0 )); then
    return 0
  else
    return 1
  fi
}
is_even 4
if [ $? -eq 0 ]; then
  echo "Even"
else
  echo "Odd"
fi
```

Problem 3: Write a Bash function factorial that takes a number as an argument and prints its factorial.
Solution:

```
#!/bin/bash
factorial() {
    n=$1
    fact=1
    for (( i=1; i<=n; i++ ))
    do
        fact=$((fact * i))
    done
    echo "Factorial of $n is $fact"
}
factorial 5
factorial 7
```

Problem 4: Write a function max_of_three that takes three numbers as arguments and prints the largest one.

Solution:

```
#!/bin/bash
max_of_three() {
    if (( $1 >= $2 && $1 >= $3 )); then
        echo "$1 is the largest"
    elif (( $2 >= $1 && $2 >= $3 )); then
        echo "$2 is the largest"
    else
        echo "$3 is the largest"
    fi
}
max_of_three 10 25 15
max_of_three 30 20 40
```

Problem 5: Write a function is_prime that checks if a number is prime.

Solution:

```
#!/bin/bash
is_prime() {
    num=$1
    if (( num <= 1 )); then
        echo "$num is not prime"
        return
    fi
    for (( i=2; i*i<=num; i++ ))
    do
        if (( num % i == 0 )); then
            echo "$num is not prime"
            return
        fi
    done
    echo "$num is prime"
}
is_prime 7
is_prime 12
```

Made with Napkin