1. Using for loop print the commands- ls, pwd, and date.

```
#!/bin/bash

for cmd in ls pwd date
do
    echo "Running $cmd command:"
    $cmd
    echo ""
done
```

2. Print the series 0 5 10 15 20..100 using loop

```
#!/bin/bash

for num in {0..100..5}
do
    echo $num
done
```

3. Please check if a student is qualified for the AIUB scholarship or not? Consider all conditions of the existing AIUB system.

```
#!/bin/bash

echo "Enter the student's name:"
read student_name

echo "Enter the student's CGPA:"
read cgpa

echo "Enter the number of credits completed by the student:"
read completed_credits

# Checking if the student qualifies for the scholarship
if (( $(awk 'BEGIN {print ('$cgpa' >= 3.75)}') )) && ((completed_credits >= 12));
```

```
then
    echo "$student_name is qualified for the AIUB scholarship."
else
    echo "$student_name is not qualified for the AIUB scholarship."
fi
```

4. Check if a given number taken from user is even or odd

```
#!/bin/bash

echo "Enter a number:"
read number

# Checking if the number is even or odd
if (( number % 2 == 0 )); then
    echo "The number $number is even."
else
    echo "The number $number is odd."
fi
```

5. Check if a given character taken from user is a vowel or not.

```
#!/bin/bash
echo "Enter a character:"
read character
# Converting the character to lowercase for case-insensitive comparison
lowercase_char=$(echo "$character" | tr '[:upper:]' '[:lower:]')

# Checking if the character is a vowel
case $lowercase_char in
    [aeiou])
        echo "The character '$character' is a vowel."
        ;;
    *)
        echo "The character '$character' is not a vowel."
        ;;
esac
```

6. Check if a user is under aged or not? (above or below 18)

```bash
#!/bin/bash

# Get the current year
current_year=$(date +"%Y")

echo "Enter the user's birth year (YYYY format):"
read birth_year

# Calculate age
age=$((current_year - birth_year))

# Checking if the user is underage or not
if (( age < 18 )); then
    echo "The user is underaged (below 18 years old)."
else
    echo "The user is not underaged (18 years old or above)."
fi
```

7. Take user name and password (hidden) as input and check again if the user can login or not.

```bash
#!/bin/bash
# Predefined username and password for demonstration purposes
username="admin"
password="admin"
echo "Enter your username:"
read input_username
echo "Enter your password:"
# Read the password as hidden input
read -s input_password

# Checking if the input credentials match the predefined ones
if [[ "$input_username" == "$username" && "$input_password" == "$password" ]];
then
    echo "Login successful."
else
    echo "Login failed."
fi
```

8. Create a simple calculator.

```
#!/bin/bash

echo "Enter the first number:"
read number1

echo "Enter the second number:"
read number2

echo "Enter the operator (+, -, *, /):"
read operator

# Performing the calculation based on the entered operator
case $operator in
    +)
        result=$((number1 + number2))
        ;;
    -)
        result=$((number1 - number2))
        ;;
    *)
        result=$((number1 * number2))
        ;;
    /)
        if [ $number2 -eq 0 ]; then
            echo "Error: Division by zero is not allowed."
            exit 1
        else
            result=$(echo "scale=2; $number1 / $number2" | bc)
        fi
        ;;
    *)
        echo "Error: Invalid operator. Please use (+, -, *, /) only."
        exit 1
        ;;
esac

echo "The result is: $result"
```

9. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n

```
#!/bin/bash

echo "Enter a number (n):"
read n


# Calculate the sum of numbers from 1 to n
sum=$((n * (n + 1) / 2))

echo "The sum of numbers from 1 to $n is: $sum"
```

10.Create a shell script file and write code to count all the even odd numbers from 1 to 30. Use while loop and all numbers should be mentioned whether it is odd or even.

```
#!/bin/bash

num=1

while [ $num -le 30 ]
do
    if [ $((num % 2)) -eq 0 ]
    then
        echo "$num is even."
    else
        echo "$num is odd."
    fi
    num=$((num + 1))
done
```

11.Write a shell script that will print the following menu, 1. Basic Mathematical Calculation 2. Compare Numbers 3. Calculate CGPA 4. Exit User can choose any option from the menu and do the operation. Option 1 should pop up another menu like, 1. Addition 2. Subtraction 3. Multiplication 4. Division Each option in the menu will have two input fields. Inputs can be integer/float. Option 2 will print the greatest and lowest numbers between the inputs (three inputs). Option 3 will take input of no of semesters and GPA for each semester and calculate the CGPA. If CGPA is less than 2.50 student is in probation (print appropriate massage). Also print if the student is up for any medal. (>=3.50 bronze medal, >=3.75 silver medal, 4.00 gold medal) Once one operation is done the main menu should pop up again. Option 4 should quit the whole operation.

```bash
#!/bin/bash

basic_math_operation() {
  echo "1. Addition"
  echo "2. Subtraction"
  echo "3. Multiplication"
  echo "4. Division"
  echo "Enter your choice:"
  read choice

  echo "Enter first number:"
  read num1
  echo "Enter second number:"
  read num2

  case $choice in
    1) echo "Result: $(awk "BEGIN { print $num1 + $num2 }")";;
    2) echo "Result: $(awk "BEGIN { print $num1 - $num2 }")";;
    3) echo "Result: $(awk "BEGIN { print $num1 * $num2 }")";;
    4) echo "Result: $(awk "BEGIN { print $num1 / $num2 }")";;
    *) echo "Invalid choice";;
  esac
}

compare_numbers() {
  echo "Enter three numbers:"
  read num1
  read num2
  read num3

  max=$(awk "BEGIN { m=$num1; if ($num2 > m) m=$num2; if ($num3 > m)
m=$num3; print m }")
  min=$(awk "BEGIN { m=$num1; if ($num2 < m) m=$num2; if ($num3 < m)
m=$num3; print m }")

  echo "Greatest number: $max"
  echo "Lowest number: $min"
}

calculate_cgpa() {
  echo "Enter the number of semesters:"
```

```
  read n

  sum=0
  for i in $(seq 1 $n)
  do
    echo "Enter CGPA for semester $i:"
    read cgpa
    sum=$(awk "BEGIN { print $sum + $cgpa }")
  done

  avg=$(awk "BEGIN { print $sum / $n }")
  echo "Overall CGPA: $avg"

  if awk "BEGIN { exit !($avg < 2.50) }"; then
    echo "Student is on probation."
  fi

  if awk "BEGIN { exit !($avg >= 4.00) }"; then
    echo "Student is awarded a Gold medal."
  elif awk "BEGIN { exit !($avg >= 3.75) }"; then
    echo "Student is awarded a Silver medal."
  elif awk "BEGIN { exit !($avg >= 3.50) }"; then
    echo "Student is awarded a Bronze medal."
  fi
}

while true
do
  echo "MENU"
  echo "1. Basic Mathematical Calculation"
  echo "2. Compare Numbers"
  echo "3. Calculate CGPA"
  echo "4. Exit"
  echo "Enter your choice:"
  read choice

  case $choice in
    1) basic_math_operation;;
    2) compare_numbers;;
    3) calculate_cgpa;;
    4) echo "Exiting..."; break;;
    *) echo "Invalid choice";;
```

```
  esac
  echo ""
done
```

12. Write a shell script to calculate the sum of 1~100.

```
#!/bin/bash

sum=0
for i in {1..100}
do
  sum=$((sum + i))
done

echo "The sum of 1 to 100 is: $sum"
```

13. Write a shell script that will ask for following inputs from the user and print them, Name, Occupation, Institution, Id No, Date. (User will not input any date. Print current date)

```
#!/bin/bash

echo "Please enter your name:"
read name

echo "Please enter your occupation:"
read occupation

echo "Please enter your institution:"
read institution

echo "Please enter your ID number:"
read id

date=$(date "+%Y-%m-%d")

echo "Name: $name"
echo "Occupation: $occupation"
echo "Institution: $institution"
echo "ID Number: $id"
echo "Date: $date"
```