Shell scripting programming examples

1. Program: Hello World

#!/bin/bash

echo "Hello, World!"

Output: Hello, World!

2. Program: Addition of Two Numbers

#!/bin/bash

num1=10

num2=20

sum=$((num1 + num2))

echo "The sum of $num1 and $num2 is $sum."

Output: The sum of 10 and 20 is 30.

3. Program: Looping through an Array

#!/bin/bash

fruits=("Apple" "Banana" "Orange" "Grapes")

echo "List of fruits:"

for fruit in "${fruits[@]}"; do

 echo "$fruit"

done

Output: List of fruits: Apple Banana Orange Grapes

4. Program: Factorial Calculation

#!/bin/bash

num=5

factorial=1

for ((i=1; i<=num; i++)); do

 factorial=$((factorial \* i))

done

echo "The factorial of $num is $factorial."

Output: The factorial of 5 is 120.

Program: Check Even or Odd

5.check Even or Odd

#!/bin/bash

num=17

if ((num % 2 == 0)); then

 echo "$num is even."

else

 echo "$num is odd."

fi

 Output: 17 is odd.

7. Program: File Backup

 #!/bin/bash

 source\_file="data.txt"

 backup\_file="data\_backup.txt"

 cp "$source\_file" "$backup\_file"

 echo "Backup created: $backup\_file"

Output: Backup created: data\_backup.txt

8. Program: Check Prime Number

#!/bin/bash

num=11

is\_prime=true

for ((i=2; i<num; i++)); do

 if ((num % i == 0)); then

 is\_prime=false

 break

 fi

done

if $is\_prime; then

 echo "$num is a prime number."

else

 echo "$num is not a prime number."

fi

Output: 11 is a prime number.

9. Program: Generate Fibonacci Sequence

#!/bin/bash

num=10

a=0

b=1

echo "Fibonacci sequence up to $num terms:"

for ((i=0; i<num; i++)); do

 echo -n "$a "

 temp=$((a + b))

 a=$b

 b=$temp

done

echo

Output: Fibonacci sequence up to 10 terms: 0 1 1 2 3 5 8 13 21 34

10. Program: File Size Calculation

 #!/bin/bash

file="example.txt"

size=$(du -h "$file" | awk '{print $1}')

echo "Size of $file is $size."

Output: Size of example.txt is 2.5K.

11. Program: Generate Random Password

#!/bin/bash

length=8

password=$(date +%s | sha256sum | base64 | head -c $length)

echo "Random password: $password"

Output: Random password: o5QtdWQz

12. Program: String Manipulation

#!/bin/bash

string="Hello, World!"

echo "Original string: $string"

uppercase=${string^^}

echo "Uppercase string: $uppercase"

lowercase=${string,,}

echo "Lowercase string: $lowercase"

substring=${string:7:5}

echo "Substring: $substring"

Output:

Original string: Hello, World!

Uppercase string: HELLO, WORLD!

Lowercase string: hello, world!

Substring: World

13. Program: Arithmetic Operations

#!/bin/bash

num1=10

num2=5

sum=$((num1 + num2))

difference=$((num1 - num2))

product=$((num1 \* num2))

quotient=$((num1 / num2))

echo "Sum: $sum"

echo "Difference: $difference"

echo "Product: $product"

echo "Quotient: $quotient"

Output:

Sum: 15

Difference: 5

Product: 50

Quotient: 2

14. Program: Check if a Directory Exists

#!/bin/bash

directory="path/to/directory"

if [ -d "$directory" ]; then

 echo "Directory $directory exists."

else

 echo "Directory $directory does not exist."

fi

Output: Directory path/to/directory exists.

# Function to check if a number is prime

is\_prime() {

 local num=$1

 local is\_prime=true

 for ((i=2; i<num; i++)); do

 if ((num % i == 0)); then

 is\_prime=false

 break

 fi

 done

 if $is\_prime; then

 echo "$num is a prime number."

 else

 echo "$num is not a prime number."

 fi

}

# Function to calculate the factorial of a number

factorial() {

 local num=$1

 local factorial=1

 for ((i=1; i<=num; i++)); do

 factorial=$((factorial \* i))

 done

 echo "The factorial of $num is $factorial."

}

# Main program

read -p "Enter a number: " number

is\_prime "$number"

factorial "$number"

In this script, we have defined two functions: is\_prime and factorial. The is\_prime function checks if a given number is prime or not, and the factorial function calculates the factorial of a given number.

The main program prompts the user to enter a number and then calls the is\_prime and factorial functions, passing the entered number as an argument. The functions perform the respective calculations and display the results.

You can modify and extend this example to include additional functions and implement various operations based on your requirements. Functions in shell scripting are a powerful way to modularize and reuse code, making scripts more organized and maintainable.

Program: Check if a File Exists

#!/bin/bash

file\_exists() {

 local filename=$1

 if [ -e "$filename" ]; then

 echo "File $filename exists."

 else

 echo "File $filename does not exist."

 fi

}

# Main program

read -p "Enter a filename: " filename

file\_exists "$filename"

2. Program: String Reversal

#!/bin/bash

reverse\_string() {

 local string=$1

 local reversed=""

 for ((i=${#string}-1; i>=0; i--)); do

 reversed="$reversed${string:i:1}"

 done

 echo "Reversed string: $reversed"

}

# Main program

read -p "Enter a string: " input\_string

reverse\_string "$input\_string"

3. Program: Check if a Number is Palindrome

#!/bin/bash

is\_palindrome() {

 local number=$1

 local reversed=0

 local original=$number

 while ((number > 0)); do

 remainder=$((number % 10))

 reversed=$((reversed \* 10 + remainder))

 number=$((number / 10))

 done

 if [ $original -eq $reversed ]; then

 echo "$original is a palindrome."

 else

 echo "$original is not a palindrome."

 fi

}

# Main program

read -p "Enter a number: " num

is\_palindrome "$num"

4. Program: Check File Permission

#!/bin/bash

check\_file\_permission() {

 local filename=$1

 if [ -r "$filename" ]; then

 echo "File $filename is readable."

 fi

 if [ -w "$filename" ]; then

 echo "File $filename is writable."

 fi

 if [ -x "$filename" ]; then

 echo "File $filename is executable."

 fi

}

# Main program

read -p "Enter a filename: " filename

check\_file\_permission "$filename"

5. Program: Calculate Power of a Number

#!/bin/bash

power() {

 local base=$1

 local exponent=$2

 local result=1

 for ((i=1; i<=exponent; i++)); do

 result=$((result \* base))

 done

 echo "$base raised to the power of $exponent is $result."

}

# Main program

read -p "Enter the base: " base

read -p "Enter the exponent: " exponent

power "$base" "$exponent"

6. Program: Check File Extension

bash

Copy code

#!/bin/bash

check\_file\_extension() {

 local filename=$1

 local extension="${filename##\*.}"

 echo "File extension: $extension"

}

# Main program

read -p "Enter a filename: " filename

check\_file\_extension "$filename"

7. Program: Generate Fibonacci Series

bash

Copy code

#!/bin/bash

generate\_fibonacci() {

 local limit=$1

 local a=0

 local b=1

 echo "Fibonacci series up to $limit terms:"

 echo -n "$a "

 for ((i=1; i<limit; i++)); do

 echo -n "$b "

 local temp=$((a + b))

 a=$b

 b=$temp

 done

 echo

}

# Main program

read -p "Enter the limit