* **Script for 50 codes in linux**

**Script:**

#!/bin/bash

# 1. Check even or odd

even\_odd() {

num=7

if (( num % 2 == 0 )); then echo "$num is Even"; else echo "$num is Odd"; fi

}

# 2. Factorial of a number

factorial() {

num=5; fact=1

for ((i=1; i<=num; i++)); do ((fact \*= i)); done

echo "Factorial of $num is $fact"

}

# 3. Prime number check

is\_prime() {

num=17; prime=1

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

if (( num % i == 0 )); then prime=0; break; fi

done

if (( prime )); then echo "$num is Prime"; else echo "$num is Not Prime"; fi

}

# 4. Fibonacci series

fibonacci() {

a=0; b=1

echo -n "Fibonacci: "

for ((i=0; i<10; i++)); do echo -n "$a "; fn=$((a + b)); a=$b; b=$fn; done

echo

}

# 5. Reverse a number

reverse\_number() {

num=1234; rev=0

while (( num > 0 )); do rem=$(( num % 10 )); rev=$(( rev \* 10 + rem )); num=$(( num / 10 )); done

echo "Reversed number: $rev"

}

# 6. Harshad number

harshad\_number() {

num=156; sum=0; temp=$num

while (( temp > 0 )); do digit=$(( temp % 10 )); sum=$(( sum + digit )); temp=$(( temp / 10 )); done

if (( num % sum == 0 )); then echo "$num is a Harshad Number"; else echo "$num is not a Harshad Number"; fi

}

# 7. Kaprekar number

kaprekar\_number() {

num=45; sq=$(( num \* num )); len=${#sq}; mid=$(( len / 2 ))

left=${sq:0:mid}; right=${sq:mid}; left=$((left == "" ? 0 : left))

if (( left + right == num )); then echo "$num is a Kaprekar Number"; else echo "$num is not a Kaprekar Number"; fi

}

# 8. GCD

gcd() {

a=48; b=18

while (( b != 0 )); do temp=$b; b=$(( a % b )); a=$temp; done

echo "GCD: $a"

}

# 9. LCM

lcm() {

a=12; b=15; prod=$((a \* b)); x=$a; y=$b

while (( y != 0 )); do temp=$y; y=$(( x % y )); x=$temp; done

echo "LCM: $((prod / x))"

}

# 10. Count digits

count\_digits() {

num=12345; count=0

while (( num > 0 )); do num=$(( num / 10 )); ((count++)); done

echo "Digits count: $count"

}

# 11. Armstrong number

armstrong() {

num=153; sum=0; temp=$num

while (( temp > 0 )); do digit=$(( temp % 10 )); sum=$(( sum + digit \*\* 3 )); temp=$(( temp / 10 )); done

[[ $sum -eq $num ]] && echo "$num is Armstrong" || echo "$num is not Armstrong"

}

# 12. Palindrome number

palindrome() {

num=121; rev=0; temp=$num

while (( temp > 0 )); do digit=$(( temp % 10 )); rev=$(( rev \* 10 + digit )); temp=$(( temp / 10 )); done

[[ $rev -eq $num ]] && echo "$num is Palindrome" || echo "$num is not Palindrome"

}

# 13. Sum of digits

sum\_of\_digits() {

num=1234; sum=0

while (( num > 0 )); do sum=$(( sum + num % 10 )); num=$(( num / 10 )); done

echo "Sum of digits: $sum"

}

# 14. Swap numbers

swap() {

a=5; b=10; echo "Before: a=$a b=$b"

temp=$a; a=$b; b=$temp

echo "After: a=$a b=$b"

}

# 15. Leap year check

leap\_year() {

year=2024

if (( (year % 4 == 0 && year % 100 != 0) || year % 400 == 0 )); then echo "$year is Leap Year"; else echo "$year is not Leap Year"; fi

}

# 16. Area of circle

area\_circle() {

r=7; pi=3.14

area=$(echo "$pi \* $r \* $r" | bc)

echo "Area of circle: $area"

}

# 17. Simple interest

simple\_interest() {

p=1000; r=5; t=2

si=$(( (p \* r \* t) / 100 ))

echo "Simple Interest: $si"

}

# 18. Power of number

power() {

base=2; exp=4; result=1

for ((i=1; i<=exp; i++)); do result=$(( result \* base )); done

echo "$base^$exp = $result"

}

# 19. ASCII value of char

ascii\_value() {

char='A'

printf "ASCII of $char is %d\n" "'${char}"

}

# 20. Decimal to Binary

dec\_to\_bin() {

num=13; bin=""

while (( num > 0 )); do bin=$(( num % 2 ))$bin; num=$(( num / 2 )); done

echo "Binary: $bin"

}

# 21. Binary to Decimal

bin\_to\_dec() {

bin=1101; dec=0; power=0

while (( bin > 0 )); do digit=$(( bin % 10 )); dec=$(( dec + digit \* (2 \*\* power) )); bin=$(( bin / 10 )); ((power++)); done

echo "Decimal: $dec"

}

# 22. Sum of even numbers

sum\_even() {

sum=0

for ((i=2; i<=20; i+=2)); do sum=$(( sum + i )); done

echo "Sum of even numbers: $sum"

}

# 23. Sum of odd numbers

sum\_odd() {

sum=0

for ((i=1; i<=19; i+=2)); do sum=$(( sum + i )); done

echo "Sum of odd numbers: $sum"

}

# 24. Number is positive, negative or zero

check\_sign() {

num=-5

if (( num > 0 )); then echo "Positive"; elif (( num < 0 )); then echo "Negative"; else echo "Zero"; fi

}

# 25. Print multiplication table

table() {

num=5

for ((i=1; i<=10; i++)); do echo "$num \* $i = $(( num \* i ))"; done

}

# 26. Perfect number

perfect\_number() {

num=28; sum=0

for ((i=1; i<num; i++)); do if (( num % i == 0 )); then sum=$(( sum + i )); fi; done

[[ $sum -eq $num ]] && echo "$num is a Perfect Number" || echo "$num is not a Perfect Number"

}

# 27. Strong number

strong\_number() {

num=145; sum=0; temp=$num

fact() { f=1; for ((i=1;i<=$1;i++)); do ((f\*=i)); done; echo $f; }

while (( temp > 0 )); do digit=$(( temp % 10 )); sum=$(( sum + $(fact $digit) )); temp=$(( temp / 10 )); done

[[ $sum -eq $num ]] && echo "$num is a Strong Number" || echo "$num is not a Strong Number"

}

# 28. Armstrong in range

armstrong\_range() {

echo "Armstrong numbers from 1 to 500:"; for ((num=1; num<=500; num++)); do temp=$num; sum=0; while (( temp > 0 )); do digit=$(( temp % 10 )); sum=$(( sum + digit \*\* 3 )); temp=$(( temp / 10 )); done; if (( sum == num )); then echo $num; fi; done

}

# 29. Prime in range

prime\_range() {

echo "Prime numbers from 1 to 50:"; for ((num=2; num<=50; num++)); do flag=1; for ((i=2; i<=num/2; i++)); do if (( num % i == 0 )); then flag=0; break; fi; done; if (( flag )); then echo -n "$num "; fi; done; echo

}

# 30. Matrix addition

matrix\_addition() {

echo "2x2 Matrix Addition Example"

A=(1 2 3 4)

B=(5 6 7 8)

for ((i=0;i<4;i++)); do sum=$(( A[i] + B[i] )); echo -n "$sum "; if (( i % 2 == 1 )); then echo; fi; done

}

# 31. Factor pairs

factor\_pairs() {

num=12; echo -n "Factors of $num: "; for ((i=1;i<=num;i++)); do if (( num % i == 0 )); then echo -n "$i "; fi; done; echo

}

# 32. Largest of 3

largest\_three() {

a=10; b=25; c=7

if (( a > b && a > c )); then echo "Largest is $a"; elif (( b > c )); then echo "Largest is $b"; else echo "Largest is $c"; fi

}

# 33. Smallest of 3

smallest\_three() {

a=10; b=25; c=7

if (( a < b && a < c )); then echo "Smallest is $a"; elif (( b < c )); then echo "Smallest is $b"; else echo "Smallest is $c"; fi

}

# 34. Check vowel

is\_vowel() {

char='e'

case $char in a|e|i|o|u) echo "$char is a vowel";; \*) echo "$char is not a vowel";; esac

}

# 35. Count vowels in a string

count\_vowels() {

str="hello world"; count=0

for ((i=0;i<${#str};i++)); do char=${str:i:1}; case $char in a|e|i|o|u) ((count++)); esac; done

echo "Vowels: $count"

}

# 36. Count characters in string

count\_characters() {

str="Bash Script"

echo "Length: ${#str}"

}

# 37. Print right triangle pattern

pattern\_triangle() {

for ((i=1; i<=5; i++)); do for ((j=1; j<=i; j++)); do echo -n "\*"; done; echo; done

}

# 38. Inverted triangle pattern

pattern\_inverted\_triangle() {

for ((i=5; i>=1; i--)); do for ((j=1; j<=i; j++)); do echo -n "\*"; done; echo; done

}

# 39. Hollow square pattern

pattern\_hollow\_square() {

n=5

for ((i=1; i<=n; i++)); do for ((j=1; j<=n; j++)); do if (( i==1 || i==n || j==1 || j==n )); then echo -n "\*"; else echo -n " "; fi; done; echo; done

}

# 40. Floyd's triangle

floyd\_triangle() {

num=1

for ((i=1; i<=5; i++)); do for ((j=1; j<=i; j++)); do echo -n "$num "; ((num++)); done; echo; done

}

# 41. Sum of natural numbers

sum\_natural() {

n=100; sum=$(( n \* (n+1) / 2 )); echo "Sum: $sum"

}

# 42. Check palindrome string

palindrome\_string() {

str="madam"

rev=$(echo $str | rev)

[[ "$str" == "$rev" ]] && echo "$str is palindrome" || echo "$str is not palindrome"

}

# 43. String to uppercase

string\_uppercase() {

str="bash script"

echo ${str^^}

}

# 44. String to lowercase

string\_lowercase() {

str="HELLO"

echo ${str,,}

}

# 45. Add digits until single digit

reduce\_digit() {

num=9875

while (( num >= 10 )); do sum=0; while (( num > 0 )); do sum=$((sum + num % 10)); num=$(( num / 10 )); done; num=$sum; done

echo "Reduced single digit: $num"

}

# 46. Count words in string

count\_words() {

str="This is a bash script"

echo $str | wc -w

}

# 47. Count lines in file (example.txt)

count\_lines() {

echo -e "Line1\nLine2\nLine3" > example.txt

echo "Lines: $(wc -l < example.txt)"

}

# 48. Create a file

touch\_file() {

touch newfile.txt

echo "File created: newfile.txt"

}

# 49. Delete a file

delete\_file() {

rm -f newfile.txt

echo "File deleted: newfile.txt"

}

# 50. Rename a file

rename\_file() {

touch oldfile.txt

mv oldfile.txt newfile.txt

echo "File renamed from oldfile.txt to newfile.txt"

}

**Note:** To execute the above programs/code we will need to call some functions.

even\_odd

factorial

is\_prime

fibonacci

reverse\_number

harshad\_number

kaprekar\_number

gcd

lcm

count\_digits

armstrong

palindrome

sum\_of\_digits

swap

leap\_year

area\_circle

simple\_interest

power

ascii\_value

dec\_to\_bin

bin\_to\_dec

sum\_even

sum\_odd

check\_sign

table

perfect\_number

strong\_number

armstrong\_range

prime\_range

matrix\_addition

factor\_pairs

largest\_three

smallest\_three

is\_vowel

count\_vowels

count\_characters

pattern\_triangle

pattern\_inverted\_triangle

pattern\_hollow\_square

floyd\_triangle

sum\_natural

palindrome\_string

string\_uppercase

string\_lowercase

reduce\_digit

count\_words

count\_lines

touch\_file

delete\_file

rename\_file