1. Write a Scala program to check two given integers, and return true if one of them is 50 or if their sum is 50.

def check\_50():Unit={

print("Enter number 1 to check : ")

var num1 = readInt()

print("Enter number 2 to check : ")

var num2 = readInt()

if (num1 == 50 || num2 == 50 || (num1+num2) == 50){

println("True")}else{

println("False")

}

}

OR

def check\_50(num1 : Int, num2: Int):Boolean = num1 == 50 || num2 == 50 || num1 + num2 == 50

1. Write a Scala program to create a new string where 'Hi' is added to the front of a given string. If the string already begins with 'Hi', return the string unchanged.

def addhi():Unit={

var str1 = "Hi"

println("Enter a string : ")

var str2 = readLine()

if (str2 == "Hi"){

println(str1)

}else{

println(str1 + " "+ str2)}

}

1. Write a Scala program to create a new string which is 3 copies of the 2 front characters of a given string. If the given string length is less than 2 return the original string.

def copy3():Unit={

println("Enter a string: ")

var str = readLine()

if (str.length < 2){

println(str)

}else{

println(str.take(2)\*3)

}

}

1. Write a Scala program to check whether a given positive number is a multiple of 3 or a multiple of 7.

def check\_num():Unit={

print("Enter a number: ")

val num1 = readInt()

if(num1 % 3 ==0 && num1 % 7 ==0){

println("It is multiple of both 3 and 7")

}else if (num1 % 3 == 0){

println("It is a multiple of 3")

}else if(num1 % 7 == 0){

println("It is a multiple of 7")

}else if(num1 % 3 ==0 && num1 % 7 ==0){

println("It is multiple of both 3 and 7")

}else{

println("It is not a multiple f 3 or 7")

}

}

1. Write a Scala program to create a new string by taking the first 3 characters of a given string and return the string with the 3 characters added at both the front and back. If the given string length is less than 3, use whatever characters are there.

def modify\_string():Unit = {

println("Enter a string : ")

var str = readLine()

var prefix = " "

if (str.length<3){

prefix = str

}else{

prefix = str.take(3)

}

println(prefix+str+prefix)

}

1. Write a Scala program to check two given integers whether either of them is in the range 100..200 inclusive.

def check\_int():Unit={

println("Enter number 1: ")

var num1 = readInt()

println("Enter number 2 : ")

var num2 = readInt()

if (num1 >= 100 && num1 <=200 ||num2 >=100 && num2 <=200){

println("True, They are in the range of 100..200")

}else{

println("They are not in the range of 100..200")

}

}

1. Write a Scala program to check whether three given integer values are in the range 20 to 50 inclusive. Return "valid" if 1 or more of them are in the said range otherwise "invalid".

def check\_int():Unit={

println("Enter number 1: ")

var num1 = readInt()

println("Enter number 2 : ")

var num2 = readInt()

println("Enter number 3 : ")

var num3 = readInt()

if (num1 >=20 && num1<= 50 || num1>= 20 && num2<=50 || num3>=20 && num3<=50){

println("Valid")

}else{

println("Invalid")

}

}

1. Write a Scala program to check whether a given character presents in a string between 2 to 4 times.

def checkCharFrequency(): Unit = {

println("Enter a string: ")

val str = readLine()

println("Enter a character to check: ")

val char = readChar()

val count = str.count(\_ == char)

if (count >= 2 && count <= 4) {

println(s"The character '$char' appears $count times, which is between 2 and 4 times.")

} else {

println(s"The character '$char' does not appear between 2 and 4 times. It appears $count times.")

}

}

OR

def checkCharFrequency(): Unit = {

println("Enter a string: ")

val str = readLine()

println("Enter a character to check: ")

val char = readChar()

var count = 0

for (ch <- str) {

if (ch == char) {

count = count + 1

}

}

if (count >= 2 && count <= 4) {

println(s"The character '$char' appears $count times, which is between 2 and 4 times.")

} else {

println(s"The character '$char' does not appear between 2 and 4 times. It appears $count times.")

}

}

1. Write a Scala program to check whether two given positive integers have the same last digit.

def checkLastDigit(): Unit = {

println("Enter the first number: ")

val num1 = readInt()

println("Enter the second number: ")

val num2 = readInt()

val lastDigit1 = num1 % 10

val lastDigit2 = num2 % 10

if (lastDigit1 == lastDigit2) {

println(s"The numbers $num1 and $num2 have the same last digit.")

} else {

println(s"The numbers $num1 and $num2 do not have the same last digit.")

}

}

1. Write a Scala program to convert the last 4 characters of a given string in upper case. If the length of the string has less than 4 then uppercase all the characters.

def convertLastFourToUpper(): Unit = {

println("Enter a string: ")

val str = readLine()

val result = if (str.length < 4) {

str.toUpperCase()

} else {

val prefix = str.substring(0, str.length - 4)

val suffix = str.substring(str.length - 4).toUpperCase()

prefix + suffix

}

println("Modified string: " + result)

}

OR

def convertLastFourToUpper(): Unit = {

println("Enter a string: ")

val str = readLine()

val result = if (str.length < 4) {

str.toUpperCase()

} else {

val lastFourUpper = str.takeRight(4).toUpperCase()

str.dropRight(4) + lastFourUpper

}

println("Modified string: " + result)

}

1. Write a Scala program to print the Fibonacci series up to a given number using a while loop.

def fibonacciSeries(): Unit = {

println("Enter a number: ")

val n = readInt()

var a = 0

var b = 1

var count = 0

println("Fibonacci series up to " + n + ":")

while (a <= n) {

println(a)

val next = a + b

a = b

b = next

count += 1

}

}

1. Write a Scala program to find the sum of all elements in an array using a for loop.

def sumOfElements(): Unit = {

println("Enter the number of elements: ")

val n = readInt()

var sum = 0

println(s"Enter $n numbers: ")

for (\_ <- 1 to n) {

val num = readInt()

sum += num

}

println("The sum of all elements is: " + sum)

}

1. Write a Scala program to print the multiplication table of a given number using a for loop.

def multiplicationTable(): Unit = {

println("Enter a number: ")

val num = readInt()

println("Enter the range till which you want the multiplication table: ")

val range = readInt()

println(s"Multiplication table of $num:")

for (i <- 1 to range) {

println(s"$num x $i = ${num \* i}")

}

}

1. Write a Scala program to check if a given string is a palindrome using if/else statements and pattern matching.

def checkPalindrome(): Unit = {

println("Enter a string: ")

var str = readLine()

var isPalindrome = true

var length = str.length

for (i <- 0 until length / 2) {

if (str(i) != str(length - i - 1)) {

isPalindrome = false

}

}

if (isPalindrome) {

println("The string is a palindrome.")

} else {

println("The string is not a palindrome.")

}

}

OR

def checkPalindrome(): Unit = {

println("Enter a string: ")

val str = readLine()

val reversedStr = str.reverse

if (str == reversedStr) {

println(s"The string '$str' is a palindrome.")

} else {

println(s"The string '$str' is not a palindrome.")

}

}

1. Write a Scala program to count the number of vowels in a given string using if/else statements and pattern matching.

def countVowels(): Unit = {

println("Enter a string: ")

val str = readLine()

var count = 0

for (char <- str) {

char match {

case 'a' | 'A' | 'e' | 'E' | 'i' | 'I' | 'o' | 'O' | 'u' | 'U' =>

count += 1

case \_ =>

}

}

println(s"The number of vowels in the string is: $count")

}

1. Write a Scala program to find the largest element in an array using pattern matching.

def findLargestElement(): Unit = {

println("Enter the number of elements in the array: ")

val n = readInt()

println("Enter the elements: ")

val arr = new Array[Int](n)

for (i <- 0 until n) {

arr(i) = readInt()

}

arr match {

case Array() => println("The array is empty.")

case arr => println(s"The largest element in the array is: ${arr.max}")

}

}

1. Write a Scala function to check if a given number is prime.

def isPrime(): Unit = {

print("Enter a number: ")

var num = readInt()

var isPrime = true

var i = 2

while (i \* i <= num && isPrime) {

if (num % i == 0) {

isPrime = false

}

i += 1

}

if (isPrime && num > 1) {

println(s"$num is prime")

} else {

println(s"$num is not prime")

}

}

1. Write a Scala function to calculate the sum of digits in a given number.

def sumOfDigits(): Unit = {

print("Enter a number: ")

var num = readInt()

var sum = 0

while (num > 0) {

sum += num % 10

num /= 10

}

println(s"Sum of digits: $sum")

}

1. Write a Scala program to convert the last 4 characters of a given string in upper case. If the length of the string has less than 4 then uppercase all the characters.

def convertToUpper(): Unit = {

println("Enter a string: ")

var str = readLine()

if (str.length < 4) {

println(str.toUpperCase())

} else {

val lastFour = str.takeRight(4).toUpperCase()

println(str.dropRight(4) + lastFour)

}

}

1. Write a Scala program to calculate the sum of the last 3 elements of an array of integers. If the array length is less than 3 then return the sum of the array. Return 0 if the array is empty.

def sumLastThreeElements(): Unit = {

println("Enter the array size: ")

val size = readInt()

var arr = Array.ofDim[Int](size)

for (i <- 0 until size) {

arr(i) = readInt()

}

val sum = if (arr.length < 3) arr.sum else arr.takeRight(3).sum

println(s"Sum of last 3 elements: $sum")

}

1. Write a Scala program to reverse a given number.

def reverseNumber(): Unit = {

println("Enter a number: ")

var num = readInt()

var reversed = 0

while (num != 0) {

reversed = reversed \* 10 + num % 10

num /= 10

}

println(s"Reversed number: $reversed")

}

1. Write a Scala program to find out if the entered year is leap year or not.

def checkLeapYear(): Unit = {

println("Enter a year: ")

val year = readInt()

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

println(s"$year is a leap year.")

} else {

println(s"$year is not a leap year.")

}

}

1. Write a Scala program to print Fibonacci series up to 100.

def fibonacciSeries(): Unit = {

var a = 0

var b = 1

println("Fibonacci series up to 100: ")

print(a + " " + b + " ")

while (a + b <= 100) {

val next = a + b

print(next + " ")

a = b

b = next

}

}

1. Write a Scala program to find whether a string is palindrome or not.

def checkPalindrome(): Unit = {

println("Enter a string: ")

var str = readLine()

var isPalindrome = true

var length = str.length

for (i <- 0 until length / 2) {

if (str(i) != str(length - i - 1)) {

isPalindrome = false

}

}

if (isPalindrome) {

println("The string is a palindrome.")

} else {

println("The string is not a palindrome.")

}

}