Practical No. 3: Programs on While Loop

# Aim: Write python programs using while loop

# 1. Write a python program using a while loop that asks the user for a number, and prints the digits of the number on screen

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Declare empty string variable

string = ""

# While num is not zero store remainder in digit,

# divide num by 10 and add digit to string

while num != 0:

digit = num % 10

num = int(num / 10)

string += " " + str(digit)

# Print reversed string

print(string[::-1])

## Output:



# 2. Write a python program to perform addition of 2 numbers till the user wishes

## Source Code:

# cont variable decides whether program should restart or terminate

cont = "yes"

while cont.casefold() == "yes":

# Get input of 2 numbers from user

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

# Add the numbers and print the answer

ans = num1 + num2

print("The sum of two numbers is", ans)

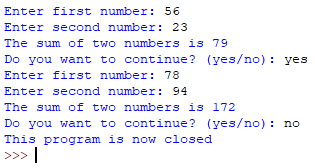
# Ask user whether they want to restart/continue the program

cont = input("Do you want to continue? (yes/no): ")

else:

print("This program is now closed")

## Output:



# 3. Write a python program to calculate factorial of a number

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Store user input num in original

original = num

# Initialize fact variable as 1

fact = 1

# Multiply num to fact and subtract 1 from num till num > 1

while num > 1:

fact \*= num

num -= 1

# Print value of original number and its factorial

print("{0}! is {1}".format(original, fact))

## Output:



# 4. Write a python program to check number is prime or not

## Source Code:

# Required for math.sqrt() function

import math

# Get input from user

num = int(input("Enter a number: "))

# Initilize divisor as 2

divisor = 2

# Till divisor < square root of num

while divisor <= int(math.sqrt(num)):

# If remainder is zero number is not prime. Break loop

if num % divisor == 0:

print("Number is not prime")

break

# Increment divisor by 1

divisor += 1

# If num % divisor is not zero number is prime

else:

print("Number is prime")

## Output:

### Case 1: Number is prime



### Case 2: Number is not prime



# 5. Write a python program to check whether number is Armstrong or not

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Initialize sumofcubes to 0

sumofcubes = 0

# Store user input num in original

original = num

while num != 0:

# Find the digit, divide num by 10

# and add cube of digit to sumofcubes

digit = num % 10

num = int(num / 10)

sumofcubes += digit \*\* 3

if sumofcubes == original:

print("This is an Armstrong number")

else:

print("This is not an Armstrong number")

## Output:

### Case 1: Number is Armstrong



### Case 2: Number is not Armstrong



# 6. Write a python program to display prime numbers between 1 to 100

## Source Code:

# Required for math.sqrt() function

import math

# Initialize num to 1

num = 1

while num <= 100:

# Initilize divisor to 2 and factor to 1

divisor = 2

factor = 0

while divisor < math.sqrt(num):

# Check if remainder is zero

if num % divisor == 0:

# Increment factor and break

factor += 1

break

# Increment divisor

divisor += 1

# Print num if prime

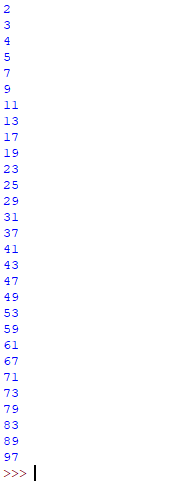
if factor == 0 and num != 1:

print(num)

# Increment num

num += 1

## Output:



# 7. Write a python program to check whether number is palindrome or not

## Source Code:

### Method 1: Using while loop

# Get input from user

num = int(input("Enter a number: "))

# Initialize rev to zero and store num in original

rev = 0

original = num

# Calculate reverse of number

while num > 0:

rem = num % 10

rev = (rev \* 10) + rem

num = int(num / 10)

# Check if original is same as reverse

if original == rev:

print("{0} is a palindrome".format(original))

else:

print("{0} is not a palindrome".format(original))

### Method 2: Using string conversion

# Get input from user

num = int(input("Enter a number: "))

# Check if string of num is same even if reversed

if str(num) == str(num)[::-1]:

print("{0} is a palindrome".format(num))

else:

print("{0} is not palindrome".format(num))

## Output:

### Case 1: Number is a palindrome



### Case 2: Number is not a palindrome



# 8. Write a python program to check whether a number is perfect or not

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Initialize additon and divisor to 0 and 1 respectively

addition = 0

divisor = 1

# Add the factors of num and store in addition

while(divisor < num):

if(num % divisor == 0):

addition += divisor

divisor += 1

# If addition is equal to the number a it is perfect number

if(addition == num):

print("{0} is a perfect number".format(num))

else:

print("{0} is not a perfect number".format(num))

## Output:

### Case 1: Number is a perfect number



### Case 2: Number is not a perfect number



# 9. Write a program to print factors of a number

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Initialize divisor to 1

divisor = 1

# Print the factors of num

print("Factors of {0} are:".format(num), end = " ")

while(divisor <= num):

if(num % divisor == 0):

print(divisor, end = " ")

divisor += 1

## Output:



# 10. Write a program to reverse a number

## Source Code:

# Get input from user

num = int(input("Enter a number: "))

# Store num in original

original = num

# Initialize reverse to zero

reverse = 0

# Calculate reverse of original number

while num != 0:

remainder = num % 10

num = int(num / 10)

reverse = (reverse \* 10) + remainder

# print reverse of the original number

print("Reverse of {0} is {1}".format(original, reverse))

## Output:

