**Experiment Number – 10**

**Title - Program to check whether the given number is palindrome by passing function name as an argument to another function.**

**Theory –**

You can call a function by using the following types of formal arguments −

* Required arguments
* Keyword arguments
* Default arguments
* Variable-length arguments

**Required arguments**

Required arguments are the arguments passed to a function in correct positional order. Here, the number of arguments in the function call should match exactly with the function definition. To call the function *printme()*, you definitely need to pass one argument, otherwise it gives a syntax error as follows

def printme( str ):

"This prints a passed string into this function"

print str

return;

# Now you can call printme function

printme()

### Keyword Arguments

A function called's arguments are linked to keyword arguments. When invoking a function with keyword arguments, the user may tell whose parameter value it is by looking at the parameter label. We can remove certain arguments or arrange them in a different order since the Python interpreter will connect the provided keywords to link the values with its parameters. Another way to use keywords to invoke the function() method is as follows:

#Python code to demonstrate the use of keyword arguments

# Defining a function

def function( n1, n2 ):

    print("number 1 is: ", n1)

    print("number 2 is: ", n2)

# Calling function and passing arguments without using keyword

print( "Without using keyword" )

function( 50, 30)

# Calling function and passing arguments using keyword

print( "With using keyword" )

function( n2 = 50, n1 = 30)

### Default Arguments

A default argument is a kind of parameter that takes as input a default value if no value is supplied for the argument when the function is called. Default arguments are demonstrated in the following instance.

# Python code to demonstrate the use of default arguments

# defining a function

def function( n1, n2 = 20 ):

    print("number 1 is: ", n1)

    print("number 2 is: ", n2)

# Calling the function and passing only one argument

print( "Passing only one argument" )

function(30)

### Variable-length Arguments

In Python, sometimes, there is a situation where we need to pass multiple numbers of arguments to the function. Such types of arguments are called **variable-length arguments**. We can declare a variable-length argument with the \* **(asterisk)** symbol.

def addition(\*numbers):

total = 0

for no in numbers:

total = total + no

print("Sum is:", total)

# 0 arguments

addition()

# 5 arguments

addition(10, 5, 2, 5, 4)

# 3 arguments

addition(78, 7, 2.5)

# Passing Function as Argument

Everything in Python is an object, even a function or a method. Hence, like variables, Python allows passing functions and methods as arguments. To pass a function as an argument to another function, write the name of the function without parenthesis in the function call statement (just like what we do with variables) and accept the reference of the function as a parameter in the called function.

def fun2():

print('This is fun2')

def fun1(x):

print('This is fun1')

x()

#passing fun2 to fun1

fun1(fun2)

Exercise –

1. Write a Python function to find the maximum of three numbers using default, required arguments.
2. Write a Python function to find the maximum of n numbers using variable length arguments.
3. Write a Python function to check whether a string is a pangram or not using keyword argument. (Pangrams are words or sentences containing every letter of the alphabet at least once.)