Lab2. Conditional Statements and Loops

Quiz 1. Write a Python program to check if a number is positive, negative or zero.(10 Points)

In [ ]:

```
# Asking for input.
number = float(input("Enter a number: "))

# Checking if the number is positive, negative, or zero.
if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```

```
The number is positive.
```

Quiz 2. Write a Python program that asks the user to input a number and then prints whether the number is even or odd. (10 Points)

In [ ]:

```
# Asking for input, again.
user_number = float(input("Enter a number: "))

# Checking even condition.
if user_number % 2 == 0:
    print("The number is even.")

# If not even, must be odd.
else:
    print("The number is odd.")
```

```
The number is even.
```

Quiz 3. Use a for loop to print all even numbers from 1 to 100.(10 Points)

In [82]:

```
# Glory of the for loop.
for num in range(1, 101):
    if num %2 == 0:
        print(num)
```

```
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100
```

Quiz 4. Write a Python program using a while loop that acts as a countdown timer, starting from 10 and counting down to 0.(10 Points)

In [81]:

```
# Establishing countdown variable.
countdown_num = 10

# The countdown loop.
while countdown_num >= 0:
    print(countdown_num)
    countdown_num -= 1
```

```
10
9
8
7
6
5
4
3
2
1
0
```

Quiz 5. Create a Python program that calculates the factorial of a given number using a while loop.(10 Points)

In [117…

```
# Defining variable for factorial calculation.
factorial_num = 4

# Sum must at least be the starting number.
sum = factorial_num

# Subtract from the factorial number until it reaches 1.
while factorial_num >= 1:

    # Breaking when we reach 1, otherwise we multiply by 0.
    if factorial_num == 1:
        break

    # The core factorial calculation.
    sum = sum * (factorial_num - 1)
    factorial_num = factorial_num - 1

print(sum)
```

```
24
```

Quiz 6. Write a Python program to find the largest number among three given numbers entered by the user. (10 Points)

In [ ]:

```python
# Asking the user kindly for their special numbers.
user_number_1 = float(input('Enter number 1'))
user_number_2 = float(input('Enter number 2'))
user_number_3 = float(input('Enter number 3'))

# If number 1 is greater than or equal to BOTH number 2 and number 3,
# then number 1 has to be the largest.
if (user_number_1 >= user_number_2) and (user_number_1 >= user_number_3):
    largest = user_number_1

# Otherwise, if number 2 is greater than or equal to BOTH number 1 and numbe
# then number 2 has to be the largest.
elif (user_number_2 >= user_number_1) and (user_number_2 >= user_number_3):
    largest = user_number_2

# If it's not number 1 and it's not number 2, then number 3 must be the larg
else:
    largest = user_number_3

print(f'The largest number is {largest}')

# This is my original code, I redid it above to use conditionals instead.
def find_largest_number(num_list):
    if len(num_list) > 3:
        return print('You tried a list of greater length than length 3.')
    return max(num_list)

find_largest_number([1,4,2])
```

```
The largest number is 4.0
```

Out[ ]:

```
4
```

Quiz 7. Write a Python program to check whether a year is a leap year or not. (10 Points)

In [ ]:

```python
# Getting the user input.
year = int(input("Input a year: "))

# Checking for the annoying invalid case.
if year <= 0:
    print("Cannot check leap year for year 0 or negative years!")
else:
    # Checking the 400-year rule first.
    if year % 400 == 0:
        print(f"{year} is a leap year.")
    # Checking the 100-year rule next.
    elif year % 100 == 0:
        print(f"{year} is NOT a leap year.")
    # Checking the normal 4-year rule last.
    elif year % 4 == 0:
        print(f"{year} is a leap year.")
    # If none of the above are triggered, it's just a regular year.
    else:
        print(f"{year} is NOT a leap year.")
```

```
1244 is a leap year.
```

Quiz 8. Create a Python script that prints the numbers from 1 to 100, but for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”. (10 Points)

In [85]:

```
# Looping over numbers 1 to 100.
for i in range(1,101):
    # First condition is the most specific.
    if i % 3 == 0 and i % 5 == 0:
        print('Fizzbuzz')
    # Second condition.
    elif i % 3 == 0:
        print('Fizz')
    # Third condition.
    elif i % 5 == 0:
        print('Buzz')
    # If none of the above, print the number.
    else:
        print(i)
```

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
Fizzbuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
Fizzbuzz
31
32
Fizz
34
Buzz
Fizz
37
38
Fizz
Buzz
41
Fizz
43
44
Fizzbuzz
46
47
Fizz
49
Buzz
Fizz
52
53
Fizz
Buzz
56
```

```
Fizz
58
59
Fizzbuzz
61
62
Fizz
64
Buzz
Fizz
67
68
Fizz
Buzz
71
Fizz
73
74
Fizzbuzz
76
77
Fizz
79
Buzz
Fizz
82
83
Fizz
Buzz
86
Fizz
88
89
Fizzbuzz
91
92
Fizz
94
Buzz
Fizz
97
98
Fizz
Buzz
```

Quiz 9. Write a Python script to calculate the square root of a number, but if the number is negative, print an error message.(10 Points)

In [ ]:

```
# Getting the user input.
user_input = float(input('Input a number:'))

# Checking for negative input.
if user_input < 0:
    # Printing the error, because imaginary numbers are fake.
    print('Cannot compute square root of negative number!')
else:
    # Computing the square root.
    sqrt = user_input**0.5
    print(f'The square root of {user_input} is {sqrt}')
```

```
Cannot compute square root of negative number!
```

Quiz 10: Using a for loop, write a Python program to print all prime numbers between 1 and 100 (10 Points)

In [ ]:

```
# This approach prioritizes readability, but it does a lot of unnecessary ma
# The idea is to generate composites, then list what is not composite.

# Initializing an empty list to send composite numbers to.
list_of_multiples = []

# The range of numbers for which we will create composite from.
# We only need to go up to 50 because the minimum multiple of 50 is 100.
for num in range(2,51):

    # This for loop will start at the selected number and create multiples u
    for mult_num in range(num, 51):
        composite_num = num * mult_num

        # I am using this if statement to avoid appending duplicates to the
        if composite_num not in list_of_multiples:
            list_of_multiples.append(composite_num)
        else:
            continue

# Now we will check membership, excluding 1 from the range because 1 is not
for prime_check in range(2,101):
    if prime_check not in list_of_multiples:
        print(prime_check)
```

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

In [116…

```python
# An alternate approach, but I don't think it's as readable.

# Compiling a list of composite numbers from 0 to 100.
composite_numbers = []

# Iterating over numbers from 0 to 100 to get the list.
for num in range(0, 101):
    # Checking for factors other than 1 and itself up to the number itself.
    for i in range(2, num):
        # Does number have multiples?
        if (num % i) == 0:
            composite_numbers.append(num)
            break

# Numbers not in the list must be prime.
for num in range(0, 101):
    if num not in composite_numbers and num > 1:
        print(num)
```

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

Bonus Question: Implement a number guessing game using a while loop where the user has to guess a randomly generated number between 1 and 100 until they get it right (20 Points)

In [ ]:

```python
# Importing the random module.
import random

# Generating a random number between 1 and 100.
rand_num = random.randint(1, 100)

# Initializing guess variable.
guess = None

# Initializing guess counter.
guess_counter = 1

# The guessing loop.
while guess != rand_num:

    # The users guess.
    guess = int(input("Guess a number between 1 and 100: "))

    # Comparing the guess to the random number.
    if guess < rand_num:
        print("Too low!")
    elif guess > rand_num:
        print("Too high!")

    # The user wins $1,000,000.
    else:
        print(f"Congratulations! The number is {rand_num}. You guessed the r

    # Incrementing the guess counter.
    guess_counter += 1
```

```
Too low!
Too high!
Too low!
Too low!
Too low!
Too high!
Too high!
Congratulations! The number is 58. You guessed the number in 8 attempts.
```

In [ ]: