Part 1

1. Evaluate the following expressions for `num1 = 10` and `num2 = 20`.

(a) `not (num1 < 1) and num2 < 10`

`True because num1 is not less than 1 and num2 is less than 10.`

(b) `not (num1 < 1) and num2 < 10 or num1 + num3 < 100`

`this expression will raise an error because num3 is not defined.`

(c) `not (num2 > 1) or num1 > num2 - 10`

`False because num2 is not greater than 1 and num1 is not greater than num2 - 10`

2. Give an appropriate if statement for each of the following (the value of `num` is not important):

a) Displays `'within range'` if `num` is between 0 and 100, inclusive.

```
if 0 <= num <= 100:
    print("within range")
```

(b) Displays `'within range'` if `num` is between 0 and 100, inclusive, and displays `'out of range'` otherwise.

```
if 0 <= num <= 100:
    print("within range")
else:
    print("out of range")
```

3. Rewrite the following if-else statements using a single if statement and elif:

```
if temperature >= 85 and humidity > 60:
    print ('muggy day today')
else:
    if temperature >= 85:
        print ('warm, but not muggy today')
    else:
        if temperature >= 65:
```

```
            print ('pleasant today')
        else:
            if temperature <= 45:
                print ('cold today')
            else:
                print ('cool today')
```

```
if temperature >= 85 and humidity > 60:
    print ('muggy day today')
elif temperature >= 85:
    print ('warm, but not muggy today')
elif temperature >= 65:
    print ('pleasant today')
elif temperature <= 45:
    print ('cold today')
else:
    print ('cool today')
```

4. Write a Python program in which:

(a) The user enters either `'A', 'B', or 'C'`. If `'A'` is entered, the program should display the word `'Apple'`; if `'B'` is entered, it displays `'Banana'`; and if 'C' is entered, it displays `'Coconut'`. Use nested if statements for this.

```
letter = input("Enter A, B, or C: ").upper()
if letter == "A":
    print("Apple")
elif letter == "B":
    print("Banana")
elif letter == "C":
    print("Coconut")
else:
    print("Invalid input")
```

b) Repeat question **(a)** using an if statement with `elif`

```
letter = input("Enter A, B, or C: ").upper()
if letter == "A":
    print("Apple")
if letter == "B":
    print("Banana")
if letter == "C":
    print("Coconut")
else:
    print("Invalid input")
```

headers instead.

(c) A student enters the number of college credits earned. If the number of credits is greater than or equal to 90, 'Senior Status' is displayed; if greater than or equal to 60, 'Junior Status' is displayed; if greater than or equal to 30, 'Sophomore Status' is displayed; else, 'Freshman Status' is displayed.

```
credits = int(input("Enter number of college credits earned: "))
if credits >= 90:
    print("Senior Status")
elif credits >= 60:
    print("Junior Status")
elif credits >= 30:
    print("Sophomore Status")
else:
    print("Freshman Status")
```

(e) The user enters a number. If the number is divisible by 3, the word 'Fizz' should be displayed; if the number is divisible by 5 the word 'Buzz' should be displayed and if the number is divisible by both 'FizzBuzz' should be displayed.

```
num = int(input("Enter a number: "))
if num % 3 == 0 and num % 5 == 0:
    print("FizzBuzz")
elif num % 3 == 0:
    print("Fizz")
elif num % 5 == 0:
    print("Buzz")
else:
    print(num)
```

5. Sam wants to store his series of car to a list. The list of a car are: (up to you). After creating a list he add some car and delete some car and at last there are still 5 cars left in his list. Additionally, he wants his car to be shuffled every time when the list is being displayed. [Hint: shuffle from random]

```
import random

cars = ["Mercedes", "BMW", "Audi", "Volkswagen", "Porsche", "Ferrari", "Lamborghini", "Aston Martin", "Jaguar", "Bentley"]
random.shuffle(cars)
print(cars)


cars.append("Mclaren")
cars.append("Bugatti
```

Part 2

1. Write a program that:

(a) Uses a loop to add up all the even numbers between 100 and 200, inclusive.

```
ans = 0
for i in range(100, 200+1, 2):
    ans = ans + i
print('The sum of even numbers between 100 and 200 is: ', ans)
```

```
==== RESTART: C:/Users/email/AppData/Local/Programs/Python/Python311/P2.1.py ===
The sum of even numbers between 100 and 200 is:  7650
```

(b) Sums a series of (positive) integers entered by the user, excluding all numbers that are greater than 100.

```
def cont():
    while True:
        sto = input('Continue adding numbers? (y/n): ')
        sto = sto.lower()
        if sto == 'y' or sto == 'n':
            return sto
        else:
            print('Invalid option')



theSum = 0


while True:
    num = int(float(input('Enter the number: ')))
    if 0 < num < 100:
        theSum = theSum + num
    sto = cont()
    if sto == 'n':
        break


print('The sum of all the positive numbers smaller than 100 that you provided is: ', theSum)
```

```
Enter the number: 1
Continue adding numbers? (y/n): s
Invalid option
Continue adding numbers? (y/n): y
Enter the number: 2
Continue adding numbers? (y/n): n
The sum of all the positive numbers smaller than 100 that you provided is:  3
```

(c) Solves Q2 but this time using an infinite loop, break and continue statements.

```
print('Welcome to number calculator\n'
      'This program will add all the numbers between 0 and 100 that you input\n'
      'Press (s) or (stop) to stop inputting.\n')
theSum = 0
userInput = 0
userInputBetween = 0
while True:
    num = input('Enter the number: ')
    if num.lower() == 's' or num.lower() == 'stop':
        break
    try:
        num = int(num)
        userInput += 1
        if 0 < num < 100:
            theSum += num
            userInputBetween += 1
        else:
            print("Number not between 0 and 100.")
            continue
    except ValueError:
        print('Invalid input detected!')


print('The stats are:\n'
      'Numbers you entered: %s\n'
      'Numbers you entered that were between 0 and 100: %s\n'
      'The sum of number you entered between 0 and 100: %s\n'
      'Thank you for using this program!' % (userInput, userInputBetween, theSum))
```

```
Welcome to number calculator
This program will add all the numbers between 0 and 100 that you input
Press (s) or (stop) to stop inputting.

Enter the number: 1
Enter the number: 1
Enter the number: 2
Enter the number: 3
Enter the number: 4
Enter the number: 5
Enter the number: 6
Enter the number: 7
Enter the number: 8
Enter the number: 9
Enter the number: 10
Enter the number: h
Invalid input detected!
Enter the number: s
The stats are:
Numbers you entered: 11
Numbers you entered that were between 0 and 100: 11
The sum of number you entered between 0 and 100: 56
Thank you for using this program!
```

(d) Prompts the user to enter any number of positive and negative integer values, then displays the number of each type that were entered.

```
def cont():
    while True:
        sto = input('Continue adding numbers? (y/n): ')
        sto = sto.lower()
        if sto == 'y' or sto == 'n':
            return sto
        else:
            print('Invalid option')



positive = 0
negative = 0


while True:
    num = int(float(input('Enter the number: ')))
    if num >= 0:
        positive = positive + 1
    else:
        negative = negative + 1
    st = cont()
    if st == 'n':
        break


print('You entered\nPositive numbers: %s\nNegative numbers: %s' % (positive, negative))
```

```
Enter the number: 12
Continue adding numbers? (y/n): y
Enter the number: -6
Continue adding numbers? (y/n): y
Enter the number: 111
Continue adding numbers? (y/n): n
You entered
Positive numbers: 2
Negative numbers: 1
```

2. The following while loop is meant to multiply a series of integers input by the user, until a sentinel value of 0 is entered. Indicate any errors in the code given. See if you can fix the program and get it running.

```
product = 1
num = input('Enter first number: ')
while num != 0:
    num = input('Enter first number: )
    product = product * num
  print('product = ', product)
```

```
product = 1
num = input('Enter first number: ')
while num != '0':
   product *= int(num)
   num = input('Enter next number: ')

print('product = ', product)
```

```
Enter first number: 1
Enter next number: 2
Enter next number: 3
Enter next number: 4
Enter next number: 5
Enter next number: 6
Enter next number: 0
product =  720
```

3. For each of the following, indicate which the definite loop is, and which an indefinite loop, explain your reasoning.

(a)

```
num = input('Enter a non-zero value:')
while num == 0:
    num = input('Enter a non-zero value: ')
```

This is an indefinite loop because there is no definite end condition specified for it to exit the loop. The user can enter any value for "num" and the loop will continue until the program is manually stopped or some other external event occurs. The condition that is being checked in the while loop "while num == 0" is true as long as the user enters 0, the loop will keep asking for input.

(b)

```
num = 0
while n < 10:
    print 2 ** n
    n = n + 1
```

This is a definite loop because there is a definite end condition specified for it to exit the loop. The loop will continue to execute as long as the value of "n" is less than 10. The value of "n" is incremented by 1 in each iteration of the loop, and once it reaches 10, the condition "n < 10" will evaluate to false, and the loop will exit. The number of iterations that the loop will execute is definite, in this case, it will execute 10 times.

Part 3

1. Create three dictionaries:

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
```

(a) Write code to concatenate these dictionaries to create a new one. Create a variable called `nums` to store the resulting dictionary. There are multiple ways to do this, however, one of the easiest is to convert each of the dictionaries items to a list (which can be added together) and pass them to the `dict()` constructor.

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}

nums = dict(list(dic1.items()) + list(dic2.items()) + list(dic3.items()))
```

(b) Write code to add a new key/value pair to the dictionary `nums:` `(7, 70)`

```
nums[7] = 70
```

(c) Write code to update the value of the item with key `3` in `nums` to `80`

```
nums[3] = 80
```

(d) Write code to remove the third item from dictionary `nums`.

```
nums.pop(3)
```

(e) Write code to sum all the items in the dictionary `nums`

```
total = sum(nums.values())
```

(f) Write code to multiply all the items in the dictionary `nums`

```
import operator
import functools

total = functools.reduce(operator.mul, nums.values())
```

(g) Write code to retrieve the maximum and minimum values in `nums`.

```
min_val = min(nums.values())
max_val = max(nums.values())
```

3. Create a dictionary named `password_lookup` that contains usernames as keys and passwords as associated string values. Make up data for five entries.

```
password_lookup = dict(user1='password1', user2='password2', user3='password3', user4='password4', user5='password5')
```

4. Write a program that creates an initially empty dictionary named `password_lookup`, prompting one-by-one for usernames and passwords (until a username of 'z' is read) entering each into the dictionary.

```
password_lookup = {}

while True:
    username = input("Enter a username (or 'z' to quit): ")
    if username == 'z':
        break
    password = input("Enter a password: ")
    password_lookup[username] = password

print(password_lookup)
```

5. Create a dictionary named `password_hint` that contains email addresses as keys, and associated values that contain both the users' "password security question," and the answer to the question. Make up data for dictionary entries.

```
password_hint = {
    'user1@example.com': ('What is your mother maiden name?', 'Kumari'),
    'user2@example.com': ('What is the name of your first pet?', 'Fool'),
    'user3@example.com': ('What is your favorite color?', 'purple'),
    'user4@example.com': ('What is the name of your high school?', 'Nepal High'),
    'user5@example.com': ('What is your favorite movie?', 'The Batman')
}
```

6. Create a dictionary named `member_table` that contains users' email addresses as keys, and answers to their password hints as the associated values, and a function that generates a temporary new password and stored in the table.

```
import random
import string

member_table = {
    'user1@example.com': 'Kumari',
    'user2@example.com': 'Fool',
    'user3@example.com': 'purple',
    'user4@example.com': 'Nepal High',
    'user5@example.com': 'The Batman'
}

def generate_temp_password(length: int) -> str:
    """
    Generates a temporary password of the specified length
    """
    return ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(length))

def update_password(email: str, password: str):
    """
    Updates the password of the user with the given email address
    """
    member_table[email] = password

temp_password = generate_temp_password(8)
update_password('user1@example.com', temp_password)
print(member_table)
```

Part 4 (Home Task)

1. The hangman game introduces many new concepts like *methods*, which are functions attached to values. You'll also need to learn about a data type called a *list*. Once you understand these concepts, it will be much easier to program Hangman.

1. You will need *random* module.
2.
3. You will need to use the concept of *list.*

z

p4 hangman.py - C:/Users/email/AppData/Local/Programs/Python/Python311/p4 hangman.py (3.11.1)

File Edit Format Run Options Window Help

```
import random

words = ['love', 'study', 'chill', 'books', 'word']
word = random.choice(words)

guessed_letters = []
lives = 6

while lives > 0:
    print(' '.join([letter if letter in guessed_letters else '_' for letter in word]))
    guess = input("Guess a letter: ").lower()
    if guess in guessed_letters:
        print("You already guessed that letter!")
    elif guess in word:
        guessed_letters.append(guess)
        if all([letter in guessed_letters for letter in word]):
            print("You won! The word was: " + word)
            break
    else:
        lives -= 1
        print("Incorrect! You have " + str(lives) + " lives left.")
    if lives == 0:
        print("You lost! The word was: " + word)
```

IDLE Shell 3.11.1

File Edit Shell Debug Options Window Help

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 202
AMD64)] on win32
Type "help", "copyright", "credits" or "license
>>>
= RESTART: C:/Users/email/AppData/Local/Prograr
_ _ _ _
Guess a letter: l
l _ _ _
Guess a letter: a
Incorrect! You have 5 lives left.
l _ _ _
Guess a letter: p
Incorrect! You have 4 lives left.
l _ _ _
Guess a letter: c
Incorrect! You have 3 lives left.
l _ _ _
Guess a letter: k
Incorrect! You have 2 lives left.
l _ _ _
Guess a letter: o
l o _ _
Guess a letter: g
Incorrect! You have 1 lives left.
l o _ _
Guess a letter: v
l o v _
Guess a letter: e
You won! The word was: love
>>>
```