

```
__name__ = "Weight Health Calculator"
__author__ = "Ryan Franks"
__copyright__ = "Copyright 11/2020, COP1500"
__version__ = "1.0.3"
__status__ = "Review"
```

```
"""Weight Health Calculator
Calculate Body Mass Index, or BMI - developed by Belgium statistician,
Adolphe Quetelet, in the 1800's.
Calculate Basic Metabolic Rate using height, weight, gender and age.
Calculate Calories to gain or lose weight based on Mifflin-StJeor Equation.
"""
```

```
import statistics
```

```
def input_num_height():
    """Ask user for height and return height."""
    height = int(input("Enter your Height in inches: "))
    # Error check input num_height is between 36 and 84 inches
    count = 1
    while height < 36 or height > 84:
        count += 1
        if not count == 4:
            print("You only get 3 tries. You are on try: ", count)
            height = int(input("Height between 36 and 84 inches: "))
        else:
            print("Too many tries. Re-attempt application from the "
                  "beginning.")
            exit()
    return height
```

```
def input_num_weight():
    """Ask user for weight and return height."""
    weight = int(input("Enter your Weight in pounds: "))
    # Error check input num_weight is between 50 and 550 lbs
    count = 1
    while weight < 50 or weight > 550:
        count += 1
        if not count == 4:
            print("You only get 3 tries. You are on try: ", count)
            weight = int(input("Weight between 50 and 550 pounds: "))
        else:
            print("Too many tries. Re-attempt application from the "
                  "beginning.")
            exit()
    return weight
```

```
def input_num_age():
    """Ask user for age and return age."""
    age = int(input("Enter your Age in years: "))
    # Error check input num_age is between 16 and 108
    count = 1
    while age < 16 or age > 108:
        count += 1
        if count < 4:
            print("You only get 3 tries. You are on try: ", count)
            age = int(input("Age between 16 and 108: "))
        else:
            print("Too many tries. Re-attempt application from the "
                  "beginning.")
            exit()
```

```

return age

def input_str_gender():
    """Ask user for gender (Male (m) or Female (f))
    and return gender.
    """
    gender = str(input("Enter your Gender. "
                       "Please use 'm' for Male and 'f' for Female: "))
    # Make str_gender lowercase
    gender = gender.lower()
    # Error check input str_gender is 'm' or 'f'
    count = 1
    while gender != "m" and gender != "f":
        count += 1
        if count < 4:
            print("You only get 3 tries. You are on try: ", count)
            gender = str(input("Use 'm' for Male and 'f' for Female: "))
            gender = gender.lower()
        else:
            print("Too many tries. Re-attempt application from the "
                  "beginning.")
            exit()
    return gender

def calculate_bmi_number(weight, height):
    """Calculate Body Mass Index (BMI) and return BMI."""
    bmi = float((weight * 703) / height ** 2)
    return bmi

def print_bmi_category(bmi):
    """Assign BMI Category using BMI number and print user category."""
    # BMI Categories
    print("\nThe BMI ranges are the following:")
    print("\tObese = 30 or Above")
    print("\tOverweight = 25.0 to 29.9")
    print("\tNormal Weight = 18.5 to 24.9")
    print("\tUnderweight = Under 18.5\n")
    # Assign user to BMI Category
    if bmi >= 30:
        print("\tYour BMI is in the Obese range at: ", "\033[1m" +
              format(bmi, ".1f") + "\033[0m", sep="")
    elif 25.0 <= bmi <= 29.9:
        print("\tYour BMI is in the Overweight range at: ", "\033[1m" +
              format(bmi, ".1f") + "\033[0m", sep="")
    elif 18.5 <= bmi <= 24.9:
        print("\tYour BMI is in the Normal range at: ", "\033[1m" +
              format(bmi, ".1f") + "\033[0m", sep="")
    else:
        print("\tYour BMI is in the Underweight range at: ", "\033[1m" +
              format(bmi, ".1f") + "\033[0m", sep="")

def calculate_num_weight_ideal(height):
    """Calculate weight needing to gain or lose by using the algebraically
    reworked BMI formula: num_weight_ideal is the ideal weight for the
    'Normal' range = (Height^2 * BMI Normal Constant 21.7)/703)
    Also, One-time calculation of BMI 'Normal' constant was found by adding
    the Normal range values of 18.5 and 24.9 and dividing by 2 to get the
    constant 21.7.
    """
    bmi_const = float(18.5 + 24.9) / 2

```

```

# Calculate ideal weight
weight_ideal = int((height ** 2 * bmi_const) / 703)
return weight_ideal

def calculate_num_weight_change(weight_ideal, weight):
    """Calculate weight change using the user's
    ideal weight versus their real weight
    and return the weight change difference.
    """
    if weight_ideal > weight:
        weight_change = weight_ideal - weight
    elif weight_ideal < weight:
        weight_change = weight - weight_ideal
    else:
        weight_change = 0
    return weight_change

def weight_adjustment(bmi, weight, height):
    """Calculate weight needed to lose or gain using BMI,
    current weight, current height and print the
    weight adjustment.
    """
    # Call function to calculate user ideal weight
    weight_ideal = calculate_num_weight_ideal(height)
    print("\nWeight adjustment needed to get into the 'Normal' range:\n")
    if 18.5 <= bmi <= 24.9:
        print("\033[1m" + "\tYour BMI is in the Normal range." + "\033[0m")
        print("\tYou do not have to make weight adjustments.")
        print("\tExiting the Weight Health Calculator.")
        exit()
    elif bmi > 24.9:
        # Call function to calculate weight change
        weight_change = calculate_num_weight_change(weight_ideal, weight)
        print("\033[1m" + "\tYou need to lose " + format(weight_change, "d"),
              "LBS." + "\033[0m", sep=" ")
    else:
        # BMI number less than 18.5
        # Call function to calculate change
        weight_change = calculate_num_weight_change(weight_ideal, weight)
        print("\033[1m" + "\tYou need to gain " + format(weight_change, "d"),
              "LBS." + "\033[0m", sep=" ")

def calculate_num_bmr(height, weight, gender, age):
    """Calculate Basic Metabolic Rate (BMR) using
    height, weight, gender, age and print BMR and
    return BMR.
    """
    print("\nBased on your height, weight, age, and gender your Basic "
          "Metabolic Rate (BMR) is calculated.")
    print("The BMR is a baseline daily calorie intake to help you lose "
          "or gain weight.\n")
    if gender == "m":
        # Mifflin-StJeor Equation for a male
        bmr = (10 * weight) + (6.25 * height) - (5 * age) + 5
    else:
        # Mifflin-StJeor Equation for a female
        bmr = (10 * weight) + (6.25 * height) - (5 * age) - 161
    print("\tYour current BMR is: ", "\033[1m" + "{:.0f}".format(bmr),
          "Calories Per Day" + "\033[0m", sep=" ")
    return int(bmr)

```

```

def calculate_week_weight_change(bmr, calories_used_from_log):
    """Using BMR and calories from calorie log input,
    calculate weight change for one week and print and
    return the number of pounds change.
    """
    # Multiply BMR by 7 for baseline calories for one week
    bmr_week = bmr * 7
    print("Your BMR for one week: ", "\033[1m", bmr_week, "\033[0m", sep="")
    calorie_gain = calories_used_from_log - bmr_week
    # Divide calorie_gain by 3500 to determine lbs change for one week (+,-)
    lbs_change_one_week = float(calorie_gain / 3500)
    print("\nLBS change for one week: ", "\033[1m",
          format(lbs_change_one_week, ".2f"), "\033[0m", sep="")
    return lbs_change_one_week

def input_calorie_log():
    """Calorie list(s) input for 7 days using food calories (+)
    and exercise calories (-).
    Print:
    Sum calories -- subtract exercise calories from food calories
    Average food calories -- mean()
    Average exercise calories -- mean()
    Return:
    Sum calories
    """
    print("\nAlong with using your BMR, the following log can help"
          " you set calorie goals.")
    print("Enter the number of calories consumed and the number of "
          "calories exercised for 7 days:\n")
    food_calorie_list = []
    exercise_calorie_list = []
    for i in range(1, 8):
        food_calories = int(input(f"\tFood calories consumed on day {i}: "))
        # Error Checking for food calories input
        count = 1
        while food_calories < 0 or food_calories > 10000:
            if count < 3:
                print("You only get 2 tries. You are on try: ", count)
                food_calories = int(input(f"\tFood calories consumed on "
                                          f"day {i} from 0 to 10000: "))
                count += 1
            else:
                print("Too many tries. Re-attempt application from the "
                      "beginning.")
                exit()
        food_calorie_list.append(food_calories)
        exercise_calories = int(input(f"\tExercise calories burned on "
                                      f"day {i}: "))
        # Error Checking for exercise calorie input
        count = 1
        while exercise_calories < 0 or exercise_calories > 3000:
            if count < 3:
                print("You only get 2 tries. You are on try: ", count)
                exercise_calories = int(input(f"\tExercise calories burned on "
                                              f"day {i} from 0 and 3000: "))
                count += 1
            else:
                print("Too many tries. Re-attempt application from the "
                      "beginning.")
                exit()
        exercise_calorie_list.append(exercise_calories)
    sum_calorie_lists = sum(food_calorie_list) - sum(exercise_calorie_list)

```

```

avg_food_calorie_list = statistics.mean(food_calorie_list)
avg_exercise_calorie_list = statistics.mean(exercise_calorie_list)
print("\nAverage per day food calories: ", "\033[1m",
      format(avg_food_calorie_list, ".0f"), "\033[0m", sep="")
print("Average per day exercise calories: ", "\033[1m",
      format(avg_exercise_calorie_list, ".0f"), "\033[0m", sep="")
print("Total calories for one week: ", "\033[1m", sum_calorie_lists,
      "\033[0m", sep="")
return sum_calorie_lists

def main():
    # Introduction
    print("\033[4m", "Weight Health Calculator", "\033[0m", sep="")
    print("Please enter the following information:\n")
    # Input requests to use in calculations
    # Call function to input height in inches
    num_height = input_num_height()
    # Call function to input weight in pounds
    num_weight = input_num_weight()
    # Call function to input age in years
    num_age = input_num_age()
    # Call function to input gender, m for Male or f for Female
    str_gender = input_str_gender()
    # Call function to calculate Body Mass Index (BMI) number
    bmi_number = calculate_bmi_number(num_weight, num_height)
    # Call function to assign and print BMI category
    print_bmi_category(bmi_number)
    # Call Function to calculate and print weight gain or lost
    weight_adjustment(bmi_number, num_weight, num_height)
    # Call function to calculate and print Basic Metabolic Rate (BMR)
    num_bmr = calculate_num_bmr(num_height, num_weight, str_gender, num_age)
    # Call function to input calories, plus and minus, for 1 week
    weekly_num_calories = input_calorie_log()
    # Call Function calculate_your_weight_change from log and bmr
    calculate_week_weight_change(num_bmr, weekly_num_calories)

# Call to main() #
main()

```