4MM013

Python Assessment 1

# Description

This assessment will allow you to demonstrate your Python programming abilities by producing solutions to the following three questions. For each question you will need to produce evidence of the code you have created, and also proof of the valid output generated by running your code. The code created should also satisfy the following requirements;

* Use suitable variable and function names which, where possible, identify the data type or function’s operation / role.
* Use comments to explain the primary operations, role, or behaviour of your code.
* Contain a “##” title with a description, your name and student number at the start of the program.

The total marks for the assessment is **15**.

# 

# Question 1

***(5 marks)***

The revised ‘Harris-Benedict’ method for calculating ‘Basal Metabolic Rate’ (BMR) is as follows:

|  |  |
| --- | --- |
| **Men** | ***BMR*** *= 88.362 + (13.397 x weight in kg) + (4.799 x height in cm) – (5.677 x age in years)* |
| **Women** | ***BMR*** *= 447.593 + (9.247 x weight in kg) + (3.098 x height in cm) – (4.330 x age in years)* |

Based on an individual's level of weekly exercise, the recommended daily kilocalorie intake to maintain current weight can be calculated via;

|  |  |  |
| --- | --- | --- |
| **Exercise**  **Category Code** | **Individual’s level of exercise** | **Calculation of daily intake required (kilocalories)** |
| **0** | Little to no exercise | *BMR x 1.2* |
| **1** | Light exercise (1–3 days per week) | *BMR x 1.375* |
| **2** | Moderate exercise (3–5 days per week) | *BMR x 1.55* |
| **3** | Heavy exercise (6–7 days per week) | *BMR x 1.725* |
| **4** | Very heavy exercise (twice per day, extra heavy workouts) | *BMR x 1.9* |

***Using this information,* *create a Py program that will prompt for an individual's details, and their level of weekly exercise, then calculate the corresponding BMR and daily required KC intake and return this information to the user.***

# Question 2

***(5 marks)***

Create a program that utilises a **recursive function** to calculate the sum of integer values between two given integers inclusive, i.e. 5, and 7 as inputs should return 18 (5+6+7). The program should ask the user for the low and high numbers to act as the initial and last values in this calculation, then find and return the sum. Your program will need to provide feedback if this is not possible due to; the input values not being integers, or the ‘low’ value being higher than the ‘high’ value.

Demonstrate this program works with:

1. The calculation of the sum of values between the first number of your student number and the last two digits of your student number when read as one number. (i.e. with a student number of 543820, the values will be ‘5’ and ‘20’.)
2. The calculation of a low value of “-12” and a high value of your height in feet (rounding up).
3. Suitable feedback for invalid input.

# Question 3

***(5 marks)***

Create a Py Program that will create a graph of y = f(x) where;

**f(x) = 3x2**

And where

**f(x) = 4x - 3**

Your graph will need to plot the values of y for the values of x between -5 and +15 for both lines.

To completely address this question you will need to;

* Correctly label the created graph with a title and labelled axes.
* Use a function to calculate the values of y for a given x value for both lines.
* Use plot markers for the graphs plotted points for both lines.
* Include a ‘grid’ on the graph.
* Present as evidence your code and the created graph with both lines visible.