BLG102E
ONLINE LAB SESSION
WEEK 3

Find Body Surface Area (BSA) with Boyd Formula (RECAP)

- Body Surface Area (BSA) measures the total surface area of the body and is used to calculate drug dosages.

- BSA can be calculated with Boyd formula as follows:

$$BSA(m^2) = 0.0003207 \times Height^{0.3} \times Weight^{\left(0.7285 - (0.0188 \times \log_{10} Weight) \right)}$$

where Height is in cm and Weight is in grams

- Write a C program that
  - Asks for height (cm) and weight (gr)
  - Calculates BSA using Boyd formula
  - Prints out the BSA

Calico Test of the BSA Task

- Calico module checks if your code works as how it is expected.
- Use of Calico:
  - python –m calico.cli bsa_tester.t --debug
  - You should not change the **bsa_tester.t** input!
  - https://calico.readthedocs.io/en/latest/tutorial.html#basics

- Revise your C program that it passes the cases of bsa_tester.t calico file.

- **In Exam**, you will be given a test file and your code will be graded accordingly.

Find the Digits of a Three Digit Number

- Find the digits of a given three digit number and print to the console.

- Digits can be found as follows:

  Rightmost digit = number % 10

  2nd rightmost digit = (number / 10) % 10

  ...

- Write a C program that

  - Asks a **three** digit number from the user.
  - Finds the all digits.
  - Prints the output in the following format:
    - "Digits of number «given number» are «1st digit», «2nd digit», «3rd digit» respectively\n"

Calico Test of the Digit Task

- Rename your C code file as digits.c
- Calico:
  - python –m calico.cli digit_tester.t
- Revise your C program that it passes the cases of digit_tester.t calico file.
  - For debugging use python –m calico.cli – debug digit_tester.t

Format Your Code with clang-format

- You can style your C code with **clang-format** tool.
- It can help you remove styling problems (i.e., indentation, broken lines, …etc)
- You can specify use of a format **style** with the following command:

  clang-format --style=Chromium input_file.c
- To get **help** for more options use the -h option:

  clang-format –h
- You can write the new version into file directly with **–i** argument.

  clang-format --style=Chromium -i input_file.c

Upload your Output on Ninova