Fundamental Computer Programming- C++ Lab(I)

LAB 2
The Game of BMI

Week 2, Fall 2022

International Bachelor Program in Informatics
College of Informatics
Yuan Ze University

Purposes

- Get familiar with the basic structure of a C++ program
- Get familiar with two of the three types of control statements
  - Sequence statements
  - Selection statements
  - Repetition statements (covered in next lab)
- Develop problem solving skills

Simple Data Types

- **Integer type**
  - **short, int, long, long long**

```
short x; int y; long z; long long u;
x=20; y=129;
```

- **Character type**
  - **char**

```
char x; char y; char z;
x = ‘g’; y=‘j’; if(x==y) x= z;
```

- **String type**
  - **string**

```
string x; string y, z;
x= “love”; y=“you”; if(x==y) x=z;
```

- **Boolean type**
  - **bool**

```
bool x, y, z;
x=true; y=false; if(x==y) z=true;
```

- **Real type**
  - **float, double**

```
float x, y; double z, u;
x=20.13; y=129; z=33.9
If(x ==y)
    u=z;
x =u;
```

Sequence Statements

```
S1;
S2;
S3:
…
Sn;
```

Selection Statements (1)

- Single Selection **if** statement

```
if (condition) {
S1;
S2;
…
Sk;
}
```

A block is defined. A set of statements contained within a pair of braces, i.e., { }, is called a block.

If there is only one statement in a block, the pair of braces can be omitted.

- Double selection **if** statement

```
if (condition) {
   S1;
   S2;
   …
   Sk;
}
else {
   ….
}
```

```
if (condition)
      S1;
else
      S2;

        ↓

condition ? S1 : S2
```

```
if (grade >= 60)
      cout << “passed”;
else
      cout << “failed”;
```

Parentheses ( ) are required.

```
cout << (grade >= 60 ? “passed” : “failed” );
                    OR
grade >= 60 ? cout << “passed” : cout << “failed”;
```

Selection Statements (2)

- In a multiple selection if statement (or called nested *if … else* statement), there is at least one else if.

```
if (condition) {
   S1;
   S2;
   …
   Sk;
}
else if (…) {
    …
}
else if (…) {
    …
}
else {
…
}
```

```
if (sGrade >= 80) {
   cout << “A”;
   countA = countA + 1;
}
else if (sGrade >= 70) {
   cout << “B”;
   countB = countB + 1;
}
else if (sGrade >= 60) {
   cout << “C”;
   countC = countC + 1;
}
else {
   cout << “F”;
   countF = countF + 1;
}
```

If (condition)

- What is a condition?
  - An expression evaluated into either true or false.
  - true in C++ has a value of 1 while false has a value of zero; however, if condition is any non-zero value, the condition is evaluated to be true.

```
if(0) cout << "0 ";
if(-1) cout << "-";
if(1) cout << “1 ";
if(2) cout << "2 " << endl;
```

The output will be -1 1 2.

Simple condition

- Expression does not contain logical operators such as && (AND), || (OR)
  - If(a)
  - if(a>b)
  - if(a<b)
  - if(!a),
  - if(a>=b)
  - if(a<=b)
  - if(a==b)
  - if (!(a<b))
  - If(cin >> aNum)

Complex condition

- Formed by two or more simple conditions using logical && (AND) and logical || (OR).
  - If(a && b)
  - If(a<5 && a >2) // Cant not be if(2<a<5)
  - If((a<5 || a>10) && a!=17)
  - If(cin >> aNum && aNum != -1)
  - if(ch!=‘g’ || ch!=‘j’ || ch < ‘a’ || ch > ‘t’)

Example

```
#include <iostream>
using namespace std;

int main(  )
{
    int number1;
    int number2;
    int number3;
    int result1;
    int result2;
    int result3;
    int result4;
    int sum;
    bool resultAvailable = true;
    cin >> number1 >> number2 >> number3;
     if(number2 == 0 || number3 == 0)
        resultAvailable  = false;
    else {
            result2 = number1 / number2;
             result3 = number2 % number3;
            result1 = number1 * number2;
            result4 = number1 + number2 + number3;
            sum = result1 + result2 + result3 + result4;
            }
    if(resultAvailable)
    cout << sum  << endl;
    else cout << "Results are not available!" <<endl;

    return 0;
}
```

LAB 2: Body Mass Index Calculator

- **Input:** Weight of a person in kilogram and Height in meter (real numbers) where Height>=0.9 and Height <=2.5; Weight >=20 and Weight >=200;
- Calculate the person’s BMI (Body Mass Index).
- **Output:** The person’s weight, height, BMI value, a BMI level statement, etc. BMI should also be a real number.

BMI Evaluation

- Body Mass Index Calculator

$$\boldsymbol{BMI = Weight/Height^2}$$

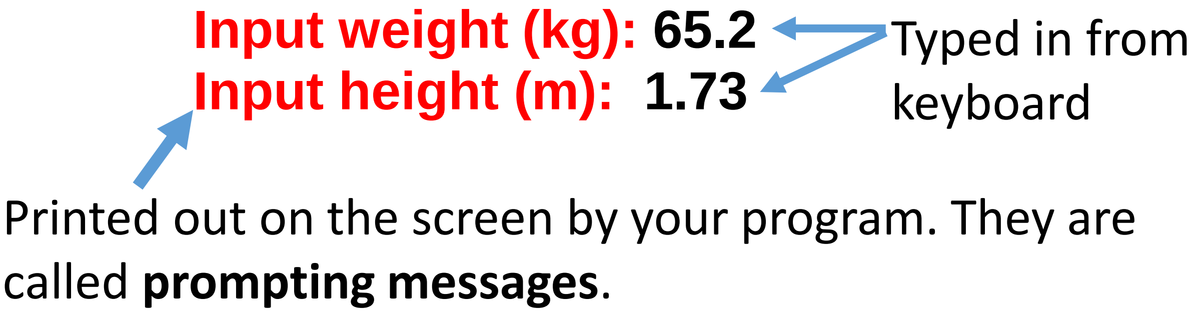
- Determine a BMI level statement

BMI level	**BMI range(kg/m^2)**
Unreasonably small	BMI <10
Highly severely underweight	10 <= BMI <= 15
Severely underweight	15 < BMI <= 16
Underweight	16 < BMI < 18.5
Normal	18.5 =< BMI <= 25
Overweight	25 < BMI <= 40
Obese	40 < BMI <= 50
Severely obese	50 < BMI <= 60
Highly severely obese	60 < BMI <= 70
Unreasonably large	BMI > 70

The Game of BMI

- **Now what you need to do is as follows:**

Step 1: get weight and height from a keyboard as shown below:



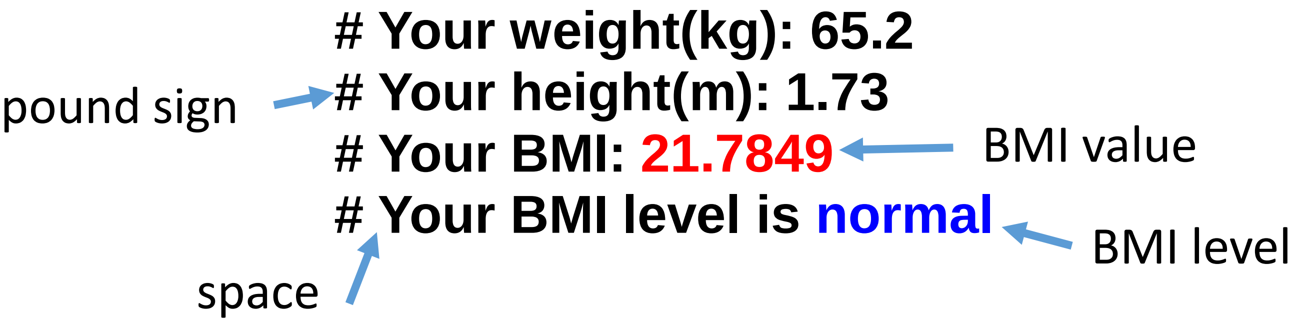
```
Input weight (kg): 65.2
Input height (m):  1.73
```

Typed in from keyboard

Printed out on the screen by your program. They are called **prompting messages**.

Step 2: check whether the input data are valid. If *Height* is invalid, then print out “# Invalid height (>= 0.9m and <= 2.5m)”. If *Weight* is invalid, then print out “# Invalid weight (>= 20kg and <= 200kg)”. Then go to Step 1 to get a new set of data. Otherwise, do Step 3.

Step 3: calculate the BMI value for the given weight and height. Then, determine the BMI level and present the following output on the monitor.



```
# Your weight(kg): 65.2
# Your height(m): 1.73
# Your BMI: 21.7849
# Your BMI level is normal
```

pound sign

BMI value

BMI level

space

The Game of BMI (2)

Step 4: if BMI value is smaller than 10, then **print out** "# Either weight or height might be wrongly given (Step 4)."

- If weight is correctly given, then calculate the largest height **H** that will make the BMI value at least 10. **Print out** "# If given weight is correct, then the given height should be at most **H** m (Step 4)."
- Similarly, if height is correctly given, then calculate the smallest weight **W** that will make BMI value at least 10. **Print out** "# If given height is correct, then the given weight should be at least **W** kg (Step 4)."

On the other hand, if BMI value is greater than 70, then **print out** "# Either weight or height might be wrongly given. (Step4)"

- If weight is correctly given, then calculate the smallest height **H** that will make the BMI value at most 70. **Print out** "# If given weight is correct, then the given height should be at least **H** m (Step 4)."
- Similarly, if height is correctly given, then calculate the largest weight **W** that will make BMI value at most 70. Print out "# If given height is correct, then the given weight should be at most **W** kg (Step 4)." Then, go to Step 1 to get a new set of data .

Otherwise, do Step 5.

$$\boldsymbol{BMI = Weight/Height^2}$$

The Game of BMI (3)

Step 5: If BMI level is not normal but its value is smaller than 18.5, then give an advice about the increase of the weight **W** to bring the BMI level into normal level. **Print out** “# Increase your weight to **W** kg to bring your BMI value to 18.5 (Step 5)“.

On the other hand if BMI level is not normal but its value is greater than 25, then given an advice about the decrease of the weight **W** to bring the BMI level to normal. **Print out** “# Decrease your weight to **W** kg to bring your BMI value to 25 (Step 5)“. Then, go to Step 1 to get a new set of data.

Grading Policy

- Step 1 to 3 takes 50%.
- Step 4 takes 25%
- Step 5 takes 25%

Constraints on Program

- You should use the following program template:

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
   double myHeight;
   double myWeight;
   double myBMI;
   bool testing = true

   while (testing) {
            .
        myBMI = round (myWeight/(myHeight*myHeight) *10)/10.0;
        Place your program here (the above line is part of your program. It is
not necessarily being the first line)
   }
   return 0;
}
```

Use this statement to calculate BMI value. The reason why it is done so just tries to deal with a problem caused by making a comparison of a floating point number to another floating point number. Comparison of two floating point numbers is sometimes very tricky. Remember what you see a floating point number on monitor may not be what is exactly stored in computer memory.

Constraints on Input Data

- Weight should be greater than or equal to 20kg and less than or equal to 200kg.
- Height should be greater than or equal to 0.9m and less than or equal to 2.5m.

Other Materials Needed

- You have to define the variables used in the program as “double” for storing real numbers. For example:

```
double myWeight;
double myHeight;
double myBMI;
```

- You may use some Boolean (logical) operators to express a condition in if (…) statement. For example:

```
If ( myWeight >= 30.3 && myHeight <= 100) {
…}
else if (myWeight <= 50 || myHeight != 200.8) {
…}
else {
 …}
```

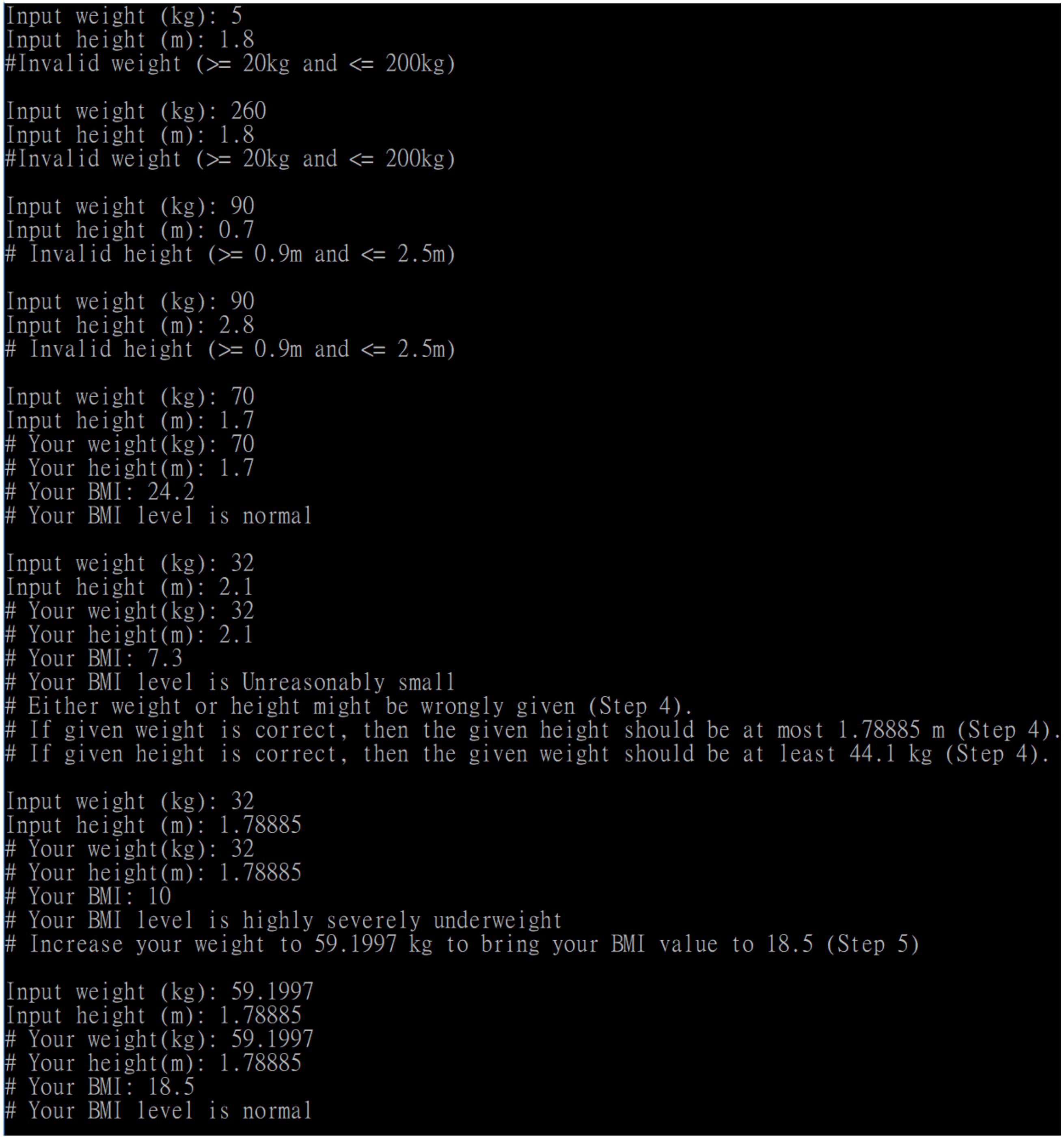
- **&& is** logical AND. X && Y is true only when X and Y are both true.
- **||** is logical OR. X || Y is true only when X or Y is true.
- Add “include <cmath>” into the program because you may use a function called sqrt to calculate the **square root** of a number.

Follow All Requirements

- Output must be exactly the same as that shown in the example
- Must use the given program template
- Must follow input format
- Must follow output format
- Must use multiple selection if (…) else if (…) to a great extent, at least eight times of else if. If not, the score is reduced by 10 points.
- Must follow the coding styles as possible as you can
  - Avoiding using variables which do not have expressive power. That is, a variable name should carry the meaning of an object in which the variable intends to represent.

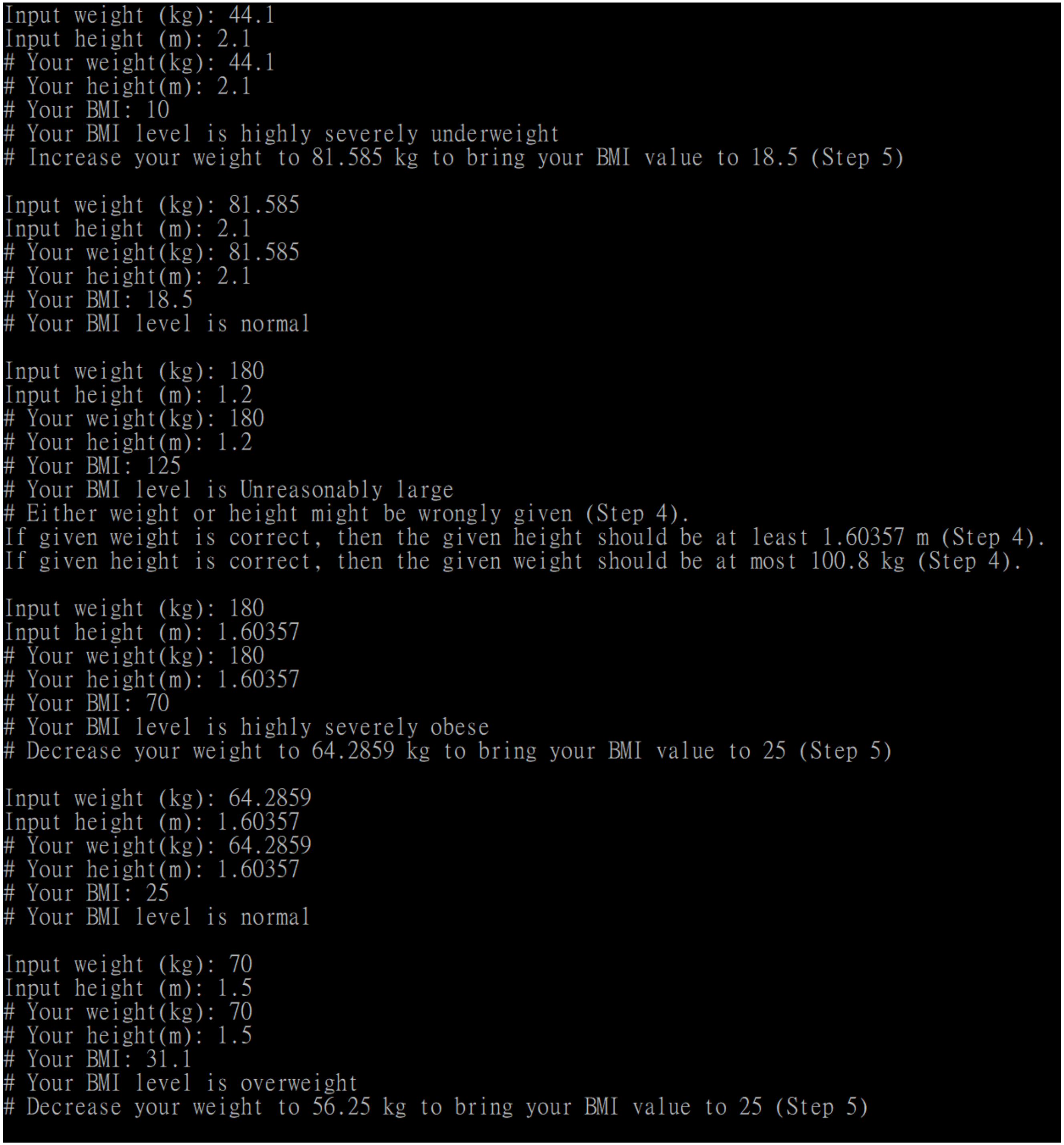
If you don't follow the requirements, up to 50% of the points for your lab will be deduced.

Example Inputs and Outputs (1)



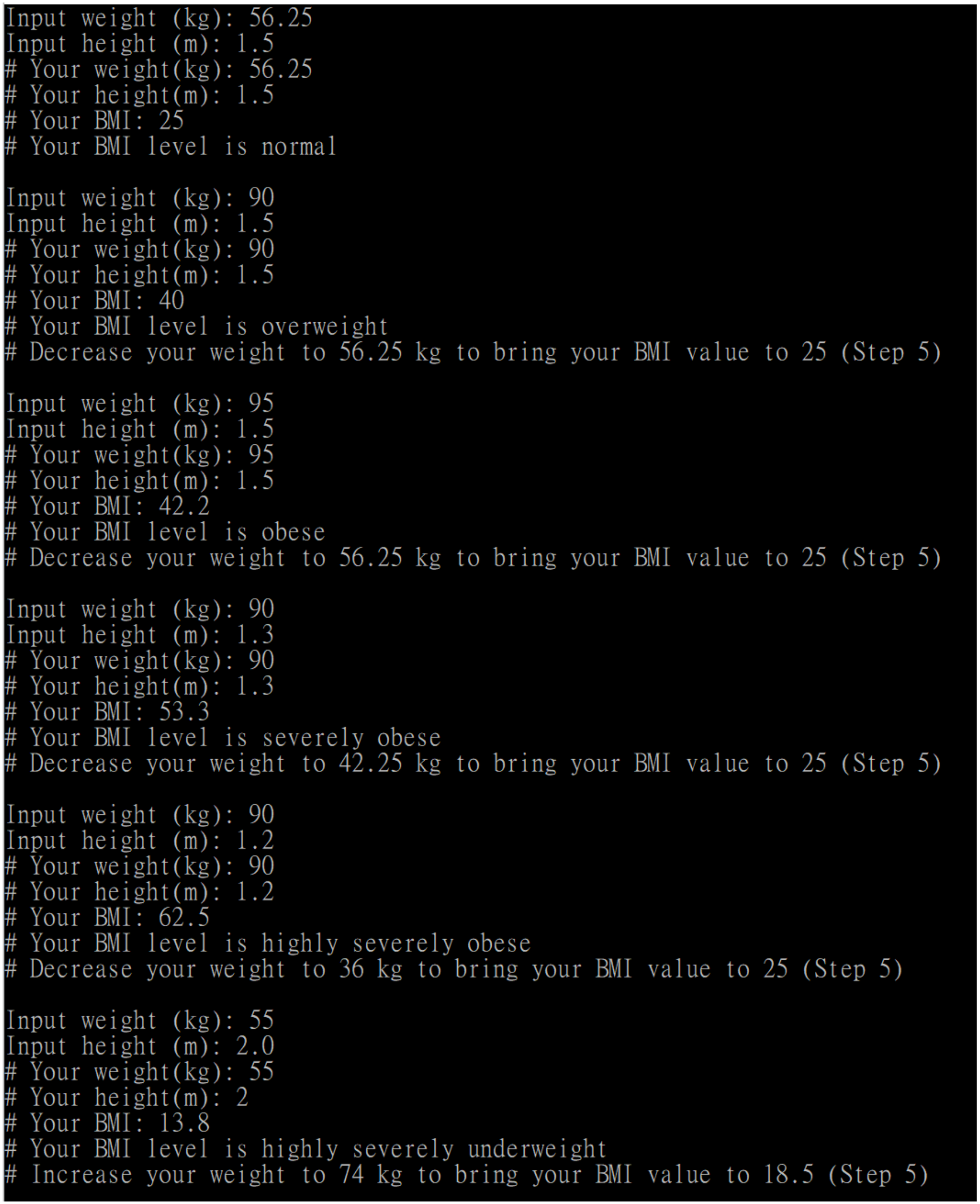
```
Input weight (kg): 5
Input height (m): 1.8
#Invalid weight (>= 20kg and <= 200kg)

Input weight (kg): 260
Input height (m): 1.8
#Invalid weight (>= 20kg and <= 200kg)

Input weight (kg): 90
Input height (m): 0.7
# Invalid height (>= 0.9m and <= 2.5m)

Input weight (kg): 90
Input height (m): 2.8
# Invalid height (>= 0.9m and <= 2.5m)

Input weight (kg): 70
Input height (m): 1.7
# Your weight(kg): 70
# Your height(m): 1.7
# Your BMI: 24.2
# Your BMI level is normal

Input weight (kg): 32
Input height (m): 2.1
# Your weight(kg): 32
# Your height(m): 2.1
# Your BMI: 7.3
# Your BMI level is Unreasonably small
# Either weight or height might be wrongly given (Step 4).
# If given weight is correct, then the given height should be at most 1.78885 m (Step 4).
# If given height is correct, then the given weight should be at least 44.1 kg (Step 4).

Input weight (kg): 32
Input height (m): 1.78885
# Your weight(kg): 32
# Your height(m): 1.78885
# Your BMI: 10
# Your BMI level is highly severely underweight
# Increase your weight to 59.1997 kg to bring your BMI value to 18.5 (Step 5)

Input weight (kg): 59.1997
Input height (m): 1.78885
# Your weight(kg): 59.1997
# Your height(m): 1.78885
# Your BMI: 18.5
# Your BMI level is normal
```

Example Inputs and Outputs (2)



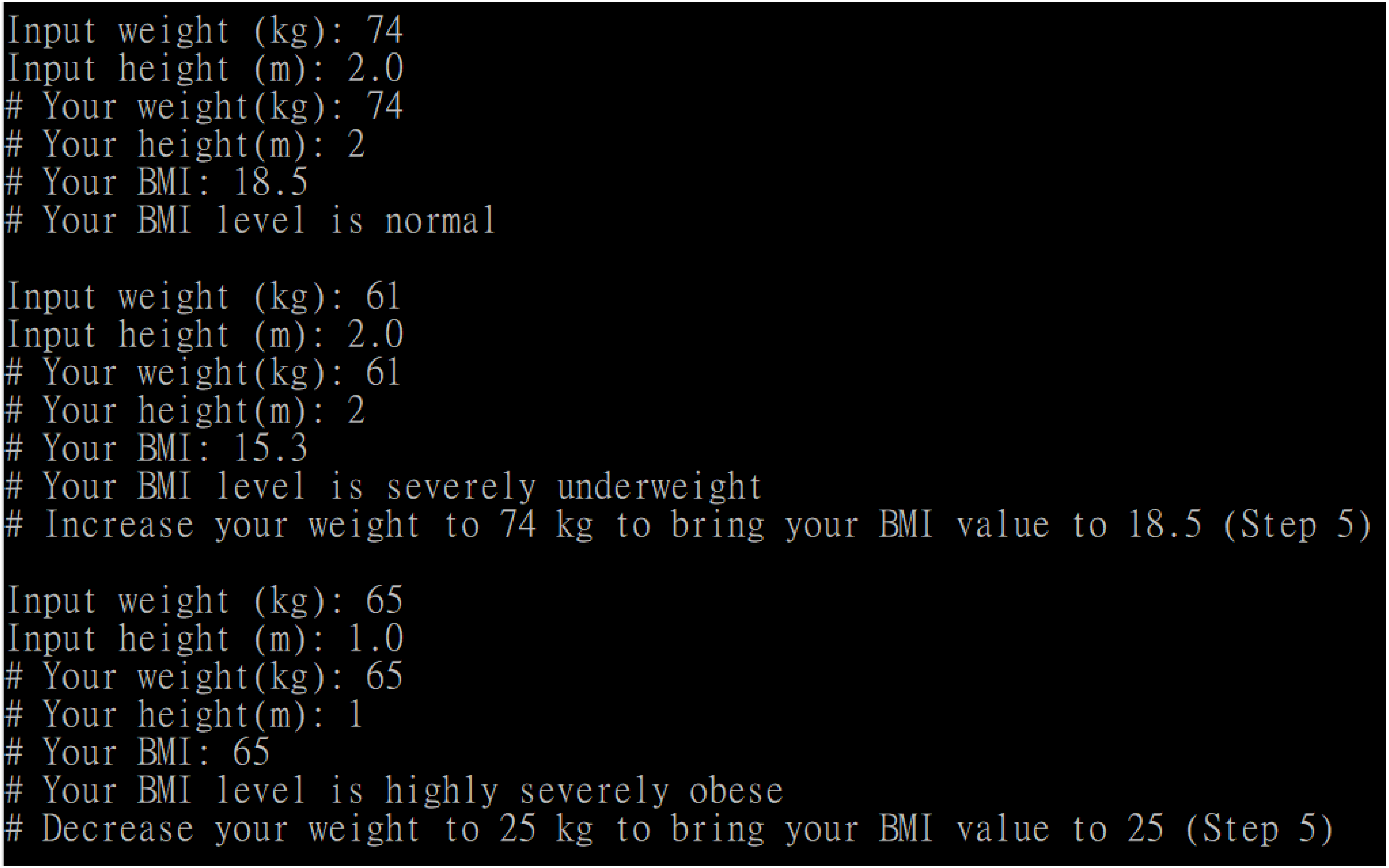
```
Input weight (kg): 44.1
Input height (m): 2.1
# Your weight(kg): 44.1
# Your height(m): 2.1
# Your BMI: 10
# Your BMI level is highly severely underweight
# Increase your weight to 81.585 kg to bring your BMI value to 18.5 (Step 5)

Input weight (kg): 81.585
Input height (m): 2.1
# Your weight(kg): 81.585
# Your height(m): 2.1
# Your BMI: 18.5
# Your BMI level is normal

Input weight (kg): 180
Input height (m): 1.2
# Your weight(kg): 180
# Your height(m): 1.2
# Your BMI: 125
# Your BMI level is Unreasonably large
# Either weight or height might be wrongly given (Step 4).
If given weight is correct, then the given height should be at least 1.60357 m (Step 4).
If given height is correct, then the given weight should be at most 100.8 kg (Step 4).

Input weight (kg): 180
Input height (m): 1.60357
# Your weight(kg): 180
# Your height(m): 1.60357
# Your BMI: 70
# Your BMI level is highly severely obese
# Decrease your weight to 64.2859 kg to bring your BMI value to 25 (Step 5)

Input weight (kg): 64.2859
Input height (m): 1.60357
# Your weight(kg): 64.2859
# Your height(m): 1.60357
# Your BMI: 25
# Your BMI level is normal

Input weight (kg): 70
Input height (m): 1.5
# Your weight(kg): 70
# Your height(m): 1.5
# Your BMI: 31.1
# Your BMI level is overweight
# Decrease your weight to 56.25 kg to bring your BMI value to 25 (Step 5)
```

Example Inputs and Outputs (3)



```
Input weight (kg): 56.25
Input height (m): 1.5
# Your weight(kg): 56.25
# Your height(m): 1.5
# Your BMI: 25
# Your BMI level is normal

Input weight (kg): 90
Input height (m): 1.5
# Your weight(kg): 90
# Your height(m): 1.5
# Your BMI: 40
# Your BMI level is overweight
# Decrease your weight to 56.25 kg to bring your BMI value to 25 (Step 5)

Input weight (kg): 95
Input height (m): 1.5
# Your weight(kg): 95
# Your height(m): 1.5
# Your BMI: 42.2
# Your BMI level is obese
# Decrease your weight to 56.25 kg to bring your BMI value to 25 (Step 5)

Input weight (kg): 90
Input height (m): 1.3
# Your weight(kg): 90
# Your height(m): 1.3
# Your BMI: 53.3
# Your BMI level is severely obese
# Decrease your weight to 42.25 kg to bring your BMI value to 25 (Step 5)

Input weight (kg): 90
Input height (m): 1.2
# Your weight(kg): 90
# Your height(m): 1.2
# Your BMI: 62.5
# Your BMI level is highly severely obese
# Decrease your weight to 36 kg to bring your BMI value to 25 (Step 5)

Input weight (kg): 55
Input height (m): 2.0
# Your weight(kg): 55
# Your height(m): 2
# Your BMI: 13.8
# Your BMI level is highly severely underweight
# Increase your weight to 74 kg to bring your BMI value to 18.5 (Step 5)
```

Example Inputs and Outputs (4)



```
Input weight (kg): 74
Input height (m): 2.0
# Your weight(kg): 74
# Your height(m): 2
# Your BMI: 18.5
# Your BMI level is normal

Input weight (kg): 61
Input height (m): 2.0
# Your weight(kg): 61
# Your height(m): 2
# Your BMI: 15.3
# Your BMI level is severely underweight
# Increase your weight to 74 kg to bring your BMI value to 18.5 (Step 5)

Input weight (kg): 65
Input height (m): 1.0
# Your weight(kg): 65
# Your height(m): 1
# Your BMI: 65
# Your BMI level is highly severely obese
# Decrease your weight to 25 kg to bring your BMI value to 25 (Step 5)
```

Rules for Program Submission

- Put all the relevant files in the same folder.
- Name your folder SID_LabX, where ID is your student ID number and X is the number assigned to the lab. If a lab has N parts, N>1, then create N sub-folders with their names SID_LabX_N in the the folder SID_LabX.
  - For example, for Lab 2 with only one part and with student ID number 1041544, the name of the folder must be S1041544_Lab2. N is omitted if there is only one part.
  - Another example, similar to the above but Lab 2 has two parts. Then, you have to create a folder S1041544_Lab2 and two sub-folders S1041544_Lab2_1 and S1041544_Lab2_2
- Compress the folder into a file named SID_LabX.zip, for example, S1041533_Lab2.zip. Then, submit the compressed file
- If you violate this rule, your lab will not be graded. If graded other penalty will be applied.