

# CS 521: Homework Assignment 2

## Assignment Directions

Complete the 5 python programming problems below. Each is worth 10 points for a total of 50 points (100%).

## Style Requirements

For all assignments, follow the guidelines in the PEP8 Standards and Best Practices that have been shared to date, along with course specific requirements.

**Remember these to avoid minor deductions:**

- Include a program docstring at the top of each program file.
- Stay under 80 characters on all code and comment lines.
- Ask for `input()` with descriptive prompts telling users what is expected.
  - There is no need to validate user input this week (that starts next week!)
- Print output that clearly explains what is being printed (where necessary). In other words, don't just print a 5 unless it's clear what that 5 represents.

***Do NOT use user defined functions for this assignment.***

## Assignment Notes

- **Some of these problems require you to take user input and others do not.** Pay careful attention to whether input is required: if the problem does not say to prompt for user input, you must not prompt for user input!
- For problems that provide sample output, take a careful look at the sample output vs. the output of running your code. Make sure your code's output matches the sample output **in any meaningful way**. For instance, if the problem requires certain outputs on a single line or in a particular order, you must present your output in the same way. However, you do not have to make your prompts or descriptions use the same wording.
  - For the sample output, any values entered by the user are shown in **red**, while output printed by the code itself are shown in **blue**.

## Submitting Your Assignment

All programs for this assignment must be submitted as *separate Python script files*.

Each program file must be named with your BU email prefix (the part before `@bu.edu`) and the problem number. For example, if your email is `alex@bu.edu`, then your first program in this assignment would be called `alex_hw_2_1.py`.

The programs must all be combined into a **single zip file** named with your email prefix and the assignment number. For example, `alex@bu.edu` would name their submission `alex_2.zip`.

# CS 521: Homework Assignment 2

## Problem 2.1

### Background

When dealing with integers, Python can easily convert between different representations — for instance, we can convert the integer `23` into binary (`10111`), octal (`27`), and hexadecimal (`17`) to make it easier to work with. (Color values in HTML, for instance, are often shown as three hexadecimal pairs representing the red, green, and blue values for a color, like `fe42a3`.)

To convert a number's format, we use some built-in helper functions:

- `int(x)` will convert `x` from a *string* to an *integer* (by default, in decimal representation)
- `bin(x)` will give you the *binary* (base-2) form of integer `x`.
- `oct(x)` will give you the *octal* (base-8) form of integer `x`.
- `hex(x)` will give you the *hexadecimal* (base-16) form of integer `x`.

### Requirements

Write a program to perform the following actions:

1. Prompt the user to enter an integer (whole number). (No validation is required for this step.)
2. Prompt the user to enter a second integer. (Again, no validation is required for this step.)
3. Multiply the first integer and the second integer.
4. Print the numbers the user entered.
5. Then print the product as a binary, octal, decimal, and hexadecimal value.

### Sample Output

```
Please enter an integer: 23
Please enter a second integer: 7
You entered the numbers 23 and 7 – their product is 161.
Here is that product in different formats:
    Binary: 0b10100001
    Octal: 0o241
    Decimal: 161
    Hexadecimal: 0xa1
```

## Problem 2.2

### Background

Consider this list of breakfast menu options in Python:

```
menu = ["ham", "eggs", "bacon", "fish", "toast", "spam", "congee", "fruit"]
```

I'd like to print out a numbered inventory of breakfast items from that list.

### Requirements

Write a program to perform the following:

1. Create the list of menu items above (copy and paste the line above into your IDE).
2. Print the total number of menu items available.
3. Using `range()` or `enumerate()`, print out a numbered list of the items on the menu (see sample output below).
  - a. Remember to make the menu item numbers start at 1 instead of 0!

## CS 521: Homework Assignment 2

### Sample Output

```
Welcome to the breakfast buffet! We have 7 items available:  
1. ham  
2. eggs  
3. bacon  
4. fish  
5. toast  
6. spam  
7. congee  
8. fruit
```

### Problem 2.3

#### Background

When dividing two numbers, we can either obtain the result as a float (e.g., `1 divided by 3 is 1.33333`) using the `/` operator. We can also obtain the result as a quotient and remainder (e.g., `1 divided by 3 is 2 remainder 1`) — the `//` operator will give us the quotient and the `%` operator will give us the remainder.

#### Requirements

Write a program to perform the following:

1. Prompt the user to enter a number. (No validation required for this problem.)
2. Prompt the user to enter a second number. (Again, no validation is required for this step.)
3. Print out the two numbers in a sentence.
4. Print out the results of dividing the **first** number by the **second** number in two ways:
  - a. First, print the result as a floating-point value, *with the result limited to two decimal places*.
  - b. Next, print the result as `X remainder Y` by using the `//` and `%` operator.

### Sample Output

```
Please enter an integer: 23  
Please enter a second integer: 7  
The result of 23 divided by 7 is:  
3.29 as a floating-point value  
3 remainder 2
```

### Problem 2.4

#### Background

Using a standard length and height, we can calculate the volume of a two-dimensional shape:

- Square: `length * length`
- Rectangle: `length * height`
- Triangle: `%(length * height)` — that is, one-half of `length` times `height`

#### Requirements

Write a program to perform the following:

1. Prompt the user to enter a value for the length of a shape.
2. Prompt the user to enter a value for the height of a shape.

## CS 521: Homework Assignment 2

3. Print back the user's input, showing the length and height values separately.
4. For the three shapes listed in the Background section above, calculate the area of the shape using the user's values (see above for formulas).
5. For each shape, print the shape name and the calculated area *with all **output** decimal values limited to two decimal places*.

### Requirement Notes

- For the input values, you may accept either integers or floats — just remember to specify in the prompt which type of input you will accept!
- Make your prompts and outputs descriptive and clear.
- Round all **output** decimal values to two digits and print all outputs as floats

### Sample Output

```
Please enter the length as a decimal (e.g., 4.0): 4.0
Please enter the height as a decimal (e.g., 4.0): 2.0
You entered a length of 4.0 and a height of 2.0.
Here are your shapes:
    Square: 16.00
    Rectangle: 8.00
    Triangle: 4.00
```

## Problem 2.5

### Background

One of the most common beginning programming problems is the fizz-buzz challenge, in which we programmatically print whether a number is divisible by other numbers. For example, if the number is divisible by three, we print "fizz"; if divisible by five, we print "buzz"; if divisible by both, we print "fizz-buzz". We're going to write a **variation on that concept** with multiple divisors and slightly different output.

### Requirements

Write a program to perform the following:

1. Declare a constant named `MAXVAL` with a value of `30`.
2. Using **either** a `for` or a `while` loop, iterate through all of the values from `1` to `MAXVAL`.
3. For each number in the loop, *on a single line*:
  - a. Print the number itself first.
  - b. If the number is divisible by 2, print the word `two`
  - c. If the number is divisible by 3, print the word `three`
  - d. If the number is divisible by 5, print the word `five`
  - e. If the number is divisible by more than one of these (e.g., divisible by two and three), *print all words that apply on the same line* (e.g., `twothree` or `twothreefive`)
    - You do not need to separate these words at all — you *may*, but you do not *need* to.
  - f. If the number is not divisible by any of these, print only the line number itself (no words) on that line (*see sample output*)
4. After the loop has completed, print a separator line.

### Requirement Notes

- Your output must present these in the correct order (`two` first, then `three`, then `five`)

## CS 521: Homework Assignment 2

- Check the sample output below for the correct way to display the output
- You are welcome to use any character (non-whitespace) value for the separator line (dashes, octothorpes, etc.)

### Sample Output

```
1:
2: two
3: three
4: two
5: five
6: twothree
7:
8: two
9: three
10: twofive
11:
12: twothree
13:
14: two
15: threefive
16: two
17:
18: twothree
19:
20: twofive
21: three
22: two
23:
24: twothree
25: five
26: two
27: three
28: two
29:
30: twothreefive
-----
```