

## Problem A. Marketing Terms [25 points]

### Note

For the following problems, write a program to solve the problem and display the answer. A possible output is shown in a example I/O section and responses to input statements appear green. Make sure you run scripts using Python 3.

### Problem

The markup of an item is the difference between its selling price and its purchase price. Two other marketing terms are

$$\text{percentage markup} = \frac{\text{markup}}{\text{purchase price}} \quad \text{and} \quad \text{profit margin} = \frac{\text{markup}}{\text{selling price}}$$

where the quotients are expressed as percentages. Write a program that computes the markup, percentage markup, and profit margin of an item. Notice that when the purchase price is tripled, the percentage markup is 200%.

### Restrictions

- Use the format method to display the following outputs.

### Example I/O

```
Enter purchase price: 250
Enter selling price: 650
Markup: $400.0
Percentage markup: 160.0%
Profit margin: 61.54%
```

### Submit format

- HW01\_A\_(NAME).py

## Problem B. Change in Salary [25 points]

### Problem

Common misconception is that if you receive three successive 10% pay raises, then your original salary will have increased by 30%. Request a salary as input and then display the salary after receiving three successive 10% pay raises. The program also should display the percentage change in salary.

### Restrictions

- Do not use any loop.
- Use the format method to display the following outputs.

### Example I/O

```
Enter beginning salary: 35000
New salary: $46,585.00
Change: 33.10%
```

### Submit format

- HW01\_B\_(NAME).py

## Problem C. Digit Sum [25 points]

### Problem

Write a program to calculate the total sum of the digits in the integers from 1 to a million.

### Restrictions

- Use only for-loop (no while-loop).
- Use the format method to display the following outputs.

### Example I/O

```
The sum of the digits in the numbers  
from 1 to one million is 27,000,001.
```

### Submit format

- HW01\_C\_(NAME).py

## Problem D. Bouncing Ball [25 points]

### Problem

The coefficient of restitution of a ball, a number between 0 and 1, specifies how much energy is conserved when the ball hits a rigid surface. A coefficient of .9, for instance, means a bouncing ball will rise to 90% of its previous height after each bounce.

Write a program to input a coefficient of restitution and an initial height in meters, and report how many times a ball bounces when dropped from its initial height before it rises to a height of less than 10 centimeters. Also report the total distance traveled by the ball before this point.

### Restrictions

- Use only while-loop (no for-loop).
- Use the format method to display the following outputs.

### Example I/O

```
Enter coefficient of restitution: .8
Enter initial height in meters: 10
Number of bounces: 21
Meters traveled: 89.08
```

### Submit format

- HW01\_D\_(NAME).py