

Name: _____

Date: ____/____/____

Midterm #1 – Additional Sample Questions

1. To make a little bit of money on the side, you decide to buy candy bars in bulk and sell them individually for a profit! In order to keep track of your revenue, you decide to write a tiny program. The program asks for the following values:
 - a) the number of boxes of candy bars
 - b) the number of individual candy bars sold
 - c) the price per candy bar sold

Once those values are entered, the program calculates your profit. Each box of 20 candy bars is \$20.

- d) calculate your profit by subtracting your expenses from your revenue.
- e) negative values for profit are acceptable (well, not really, in that you've lost money, but ok for how your program works).
- f) you do not have to check if the number of bars sold are greater than the number of bars bought in bulk. Print out the profit by saying: "After selling [number of candy bars], your total profit is [profit].".
- g) Example interaction below:

```
How many boxes of candy bars did you buy?
> 2
How many bars of candy did you sell?
> 32
How much did you sell each bar for?
> 2
After selling 32 candy bars, your total profit is $24.
```

2. Write a program that generates a random fortune.
 - a) Ask the user if they want to see the future.
 - b) If the answer YES, yes, or Yes, then generate a random number from 1 up-to and including 3.
 - c) Create 3 fortunes, and based on the number that is generated, output one of the fortunes that you created.
 - d) If the answer isn't any of the permutation of "yes" specified above, say "I didn't want to tell you anyway"
 - e) Example Output (consider text after the ">" user input):

Run 1:
Would you like to see your future? > YES
You'll have a pastry today.

Run 3:
Would you like to see your future? > no
I didn't want to tell you anyway

Run 2:
Would you like to see your future? > yes
There are many quotation marks in your future!

3. What is the output of the following code? Error is possible.

STATEMENTS	OUTPUT
<code>print(format(12, '.5f'))</code>	
<code>print(format(12000, '>,5d'))</code>	
<code>print(format(1200, '>8,d'))</code>	
<code>print("hi", "hey", sep=".")</code>	
<code>print("hi hey", end="!")</code>	

4. Assume the days of the week are numbered 1,2,3,4,5,6,7 from either Sunday to Saturday or Monday to Sunday (depending on what day is chosen to start with). Write a program that asks the user for a number and start; output the corresponding day.
- Ask the user "Start with Sun or Mon" ... if the user enters something other than "Sun" or "Mon", start from Sunday (that is, Sunday is 1, Saturday is 7). However, if the user enters "Mon", start from Monday (Monday is 1, Sunday is 7)
 - Then ask the user to enter a number: "Enter day number"
 - If the input is not between 1 and 7 inclusive, continually ask for a number until they put in a valid number ("Please enter 0-6")
 - (You don't have to worry about non-numeric input, though)
 - Example Output - Everything after the greater than sign (>) is user input:

Example 1 (Mon is 1st day)

```
Start with Sun or Mon?
> Mon
Enter day number
> 4
4 is Thursday
```

Example 2 (default start is Sun)

```
Start with Sun or Mon?
> asdfa
Enter day number
> 4
4 is Wednesday
```

Example 3 (ask again if invalid number)

```
Start with Sun or Mon?
> Sun
Enter day number
> 100
Please enter 0-6!
> -12
Please enter 0-6!
> 5
5 is Thursday
```

5. Draw a box using uppercase O's. Ask the user for the height and width of the box. Draw the box accordingly (count the top and bottom in the height, count the sides for the width). The minimum width is 3 and the minimum height is 3.
- a) ask for height
 - b) ask for width
 - c) draw a rectangle
 - d) use a for loop to deal with the height
 - e) if the width entered is less than 3, default to a width of 3
 - f) if the height entered is less than 3, default to a height of 3
 - g) hint: you can multiply strings, even if they are spaces! for example `' '* 3` will give you `' '`
 - h) hint: one way to deal with the borders is to subtract the outer border from the passed in width... but just concatenate the outer borders when your print to get `"O O"`

Example Output (text after > is user input)

```
Run 1:
-----
Please specify a height
> 3
Please specify a width
> 3
OOO
O O
OOO
```

```
Run 2:
-----
Please specify a height
> 5
Please specify a width
> 10
OOOOOOOOOO
O          O
O          O
O          O
OOOOOOOOOO
```

```
Run 3:
-----
Please specify a height
> 1
Please specify a width
> 2
OOO
O O
OOO
```

6. Describe the following *escape sequences*:

"\\" _____
"\n" _____
"\" " _____

7. Roll three three-sided dice 1000 times. Keep track of the number of each potential result (3 - 9).

- a) create a histogram showing the results
- b) you can count by 10's using an = to represent each 10 in the histogram...
- c) you can use division, the built in round() function, and int to convert from
- d) float to int to get the number of ='s to do this
- e) for example: round(2.5) returns a float... use int to change it to an int:
- f) int(round(2.5))... and finally use that to multiply your string

```
3 ====
4 =====
5 =====
6 =====
7 =====
8 =====
9 ===
```

8. Write a program that **asks the user for a number**. The program will take the number and **print out all factors of that number**. Use the algorithm below.
- Check every possible factor** of that number by counting from **one** up to and **including the number entered**.
 - If the original number entered is divisible by the factor (that is, if the remainder is 0), consider it a valid factor and print it.

Example Output:

```
Please enter a number
> 21
1
3
7
21
```

9. Write a program that continually adds exclamation points to a word as long as a user types in an uppercase or lowercase y
- ask for a word
 - then... repeatedly ask if the user wants to add an exclamation point
 - if the user types in y or Y, add an exclamation point to the string
 - continue to ask for exclamation points if the answer is y or Y
 - after every answer, print out the new version of the word with exclamation points added
 - use a `*while*` loop to do this

Run 1:

```
-----
Please enter a word
> foo
Want to add an exclamation point?
> y
foo!
Want to add an exclamation point?
> Y
foo!!
Want to add an exclamation point?
> nope
foo!!
```

Run 2:

```
-----
Please enter a word
> foo
Want to add an exclamation point?
> no
foo
```