Coding Questions:

1. Write a program that creates a dynamic dictionary which keeps track of the quantity of each item brought by all participants.

The CSV file contains equipment brought by individuals. Each line represents one person, and each cell of each line represents the item they have brought.

Your program should add to the dictionary if the dictionary does not already have the equipment, but if the dictionary already contains the equipment, you should update its quantity.

Your program should be robust to capitalization and remove leading and trailing spaces. For example ("chiPs" and " CHips" should all the counted in the same total)

Do not hardcode things that depends on the values in the cells remaining the same, as a different CSV file with the same structure will be used to test your code.

At the end the program should output the all the items participants have brough and their quantity. You should have the following output given the equipment.csv file:

Sample Output

2. Write a program that takes 2 strings from the user, a word and a substring. If the substring can be found in the word, the program should return all index positions of where the substring is found. The program should return the starting and end index in word where the substring is found. Otherwise the program will print that the substring cannot be found in word. (Hint: utilize slicing for shorter code as well as review purposes)

Example:

```
Please input a word: This is a long string and I want to know how many
    times long is in this long long sentence.
Please input a substring which you want to find in the word: long
The substring is found on index 10 to 13 of the word
The substring is found on index 56 to 59 of the word
The substring is found on index 72 to 75 of the word
The substring is found on index 77 to 80 of the word
>
```

```
Please input a word: this sentence has all the letters of the alphabet
Please input a substring which you want to find in the word: z
No instances of z found in: this sentence has all the letters of the
    alphabet
>
```

3. Write a program that takes integer inputs from the user, as many times as the user specifies in the beginning. Keep a track of the points earned by the user using the following rules. Do not print the rules to the user. Limit the input to no greater than 999.

1.Should the number be a multiple of 4, the user gets 1 point.

2.Should the number be a multiple of 7, the user gets 7 points

3.Should be number be a multiple of 5, the user would get 2 points, this rule takes precedence over the first rule and nullifies, meaning that a number such as 20 would results in 2 points.

4.Should the number be an even number, the user gets 1 point. However, if it is an even number but at the same time a multiple of 6, the user gets -2 points instead.

5.Should be number be a 2 digit number, the user would get -5 points.

6.Should the number be a multiple of 28, the user would get 100 points. This is known as a lucky guess. This rule takes precedence over all other rules, and prevents any other rules from being applied.

Unless specified otherwise, these conditions are not mutually exclusive, meaning a number could qualify any number of rules. For example if the number is 24, then the point gained is given by(1+(-2)+(-5)) which would be -6 points. This is because 24 is a multiple of 4, an even number but a multiple of 6, and a 2 digits number. The number 400 would yield 3 points, 2 for being a multiple of 5, 1 for being an even number, and none for being a multiple of 4 because the rule of 5 takes precedence.

At the end the total points earned by the user is outputted along with how many lucky guess they had. Should the user score less than 0 points, 0 points is shown. It is recommended you print out the running total to make sure your code is correct when testing.

Sample Outputs:

```
Please enter how many numbers you would like to input: 4
Please enter an integer no larger than 999: 1000
You forfeit 1 chance please enter an integer no larger than 999
Please enter an integer no larger than 999: 20000
You forfeit 1 chance please enter an integer no larger than 999
Please enter an integer no larger than 999: 28
Please enter an integer no larger than 999: 56
your score is: 200 and you had 2 lucky guesses!
>
```

```
Please enter how many numbers you would like to input: 3
Please enter an integer no larger than 999: 28
Please enter an integer no larger than 999: 20
Please enter an integer no larger than 999: 7
your score is: 105 and you had 1 lucky guesses!
>
```

```
Please enter how many numbers you would like to input: 3
Please enter an integer no larger than 999: 66
Please enter an integer no larger than 999: 23
Please enter an integer no larger than 999: 99999
You forfeit your last chance
your score is: 0
>
```