Project 1: Comparing Phones

- Project starts on: **Saturday, September 12th**
- Project is due on: **Sunday, September 27th 23:59**
- This is an individual project

Please read the entire handout before you start coding

Introduction

Most of you have had faced the dilemma of choosing a new cell phone from a variety of options. Many are the aspects you consider in your decision, but because of the number of features supported by modern cell phones the simple choice of a new phone becomes sometimes a tedious task. In order to make your decision process easier, you might apply your programming skills learned so far and write a program in Python that determine the best phone out of two based on some preset criteria.

Until now, we have mostly worked with computer programs where instructions are executed sequentially, namely, instructions of the program are executed one after another. Sequencing is fundamental in programming but not enough to solve a number of problems such as the selection of the best phone out of two. For problems like this sometimes it is required to alter the sequential flow of the program to achieve a particular goal. In Python this is done with special statements know as control structures.

In this project your will have the opportunity to work with one type of control structure. Yo will work with decision structures, which are statements that allow a program to execute different sequences of instructions for different cases. Specifically, this project require you to use the if statement (if-elif-else), one of the most common decision structures.

The challenge to you consist of reading from the user certain feature information of two different phones and calculate the value of each based on these features. Once you have the estimated values of both phones your following task is determining which of them is better based on some criteria provided in a later section.

Be aware that the completion of this project can be achieved with a correct application of the if-elif-else statement and functions that return a value. We encourage you to spend sometime thinking about the logic of your solution before start implementing it.

Setting up the environment

Go to your working directory in “data.cs.purdue.edu” and create a directory “cs177/project1”. Refer to the lab1 if you need to remember the steps to do so.
Useful material

The following material may help you prepare for the project.

Nested-if

Functions

Project skeleton

Your project skeleton is here. Download this file and start working on it!

Part 1: Computing the value of phone features

Here you need to define the function `computeValue`, which will compute and return the value of a phone based on whether the phone has GPS, has a camera, has WiFi and the condition of the phone. This information will be obtained from the user (that will be your task in the next section). The table below shows the options that can be entered by users for each feature and their respective value in parentheses.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Option 1</th>
<th>Option 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>New (+10)</td>
<td>Used (+9)</td>
</tr>
<tr>
<td>GPS</td>
<td>True (+1)</td>
<td>False (0)</td>
</tr>
<tr>
<td>WiFi</td>
<td>True (+1)</td>
<td>False (0)</td>
</tr>
<tr>
<td>Camera</td>
<td>True (+1)</td>
<td>False (0)</td>
</tr>
</tbody>
</table>

Your function definition must contain these features as parameters. You do NOT have to wait until implementing your main function (next section) to test your code. You might just call your function with the arguments/values you want to test your function. A snippet of your function definition and the way to test it is showed below.

```python
# Calculate the value of the phone based on its features and condition
# condition: state of the phone, type: string
# gps: whether the phone has GPS or not, type: boolean
# wifi: whether the phone has WiFi or not, type: boolean
# camera: whether the phone has camera or not, type: boolean
# RETURN: computed value of the phone, type: integer
def computeValue(condition, gps, wifi, camera):

    # Declare and initialize variable (integer) to accumulate value of the phone
    conditionValue = 0

    # Increment value of variable based on its features
    if condition == 'New':
        conditionValue += 10
    elif condition == 'Used':
        conditionValue += 9
    else:
        print('Invalid condition')

    if gps:
        conditionValue += 1
    else:
        conditionValue -= 1

    if wifi:
        conditionValue += 1
    else:
        conditionValue -= 1

    if camera:
        conditionValue += 1
    else:
        conditionValue -= 1

    return conditionValue
```

Note: just
to test your function

**Hint:** Define a variable in your function which accumulates the value according to the table.

### Part 2: The main function

Your second task is to write the main function. Your main function should look like as showed below. Please incrementally fill in the required missing parts.

```python
def main():
    # Declare and initialize variable (integer) to determine the best phone
    # Read feature information of phone 1
    # Read feature information of phone 2
    # Get value of phone 1
    # Get value of phone 2
    # Assign a value to the variable according the rules
    # Print final result based on value of the variable
```

As it was mentioned before, in the main function you are required to ask for the feature information of both cell phones. Part of this information is used to compute the values of the phones. Specifically, you will ask for the following user inputs.

1. What Brand is the phone?
2. What is the condition of the phone?
3. Does the phone have a camera?
4. Does the phone have GPS?
5. Does the phone have WiFi?
6. What is the price of the phone?

You need to check here that the price of the phone is neither less than 0 nor greater than 1000. If that is the case, your program should output “Invalid input, please re-run the simulation”.

### Best Phone Determining Rules

Please notice above the input validation you are expected to do. Once the user enters the required/validated values, we use the `computeValue` function to compute the values of both phones.
(Hint: Call the function twice and assign the returned value to two different variables). After having the values of both phones we proceed to evaluate which one is better based on the following rules:

1) If value of phone 1 is greater than value of phone 2 and the price of both phones is equal, then phone 1 is better.
2) If value of phone 2 is greater than value of phone 1 and the price of both phones is equal, then phone 2 is better.
3) If value of both phones is equal but price of phone 1 is higher than price of phone 2, then phone 2 is better.
4) If value of both phones is equal but price of phone 2 is higher than price of phone 1, then phone 1 is better.
5) Otherwise, it is tie.

Output Examples

Your program should clearly display which phone is better. In order to give you a clear idea of the expectations of this project find below some examples of the interaction with the user. In this examples we are going to consider two different phones Samsung and Nokia. Just to be complete we will vary their features to see all the possible expected results.

Case 1: Samsung is better than Nokia

============= PHONE ONE INPUT =============
1). What brand is your phone? Samsung
2). What is the condition of your phone? New
3). Does the phone have Camera? True
4). Does the phone have GPS? False
5). Does the phone have WiFi? False
6). Enter phone price($0-1000) : 800

============= PHONE TWO INPUT =============
1). What brand is your phone? Nokia
2). What is the condition of your phone? Used
3). Does the phone have Camera? True
4). Does the phone have GPS? False
5). Does the phone have WiFi? True
6). Enter phone price($0-1000) : 900

Samsung is better than Nokia!

In this case we get the following values for each phone:

Samsung = 10 (new) + 1 (has camera) + 0 (does not have GPS) + 0 (does not have Wi-Fi) = 11

Nokia = 9 (used) + 1 (has camera) + 0 (does not have GPS) + 1 (does not have Wi-Fi) = 11

Both cell phones have the same value but they have different prices. Based on the rules defined above (specifically point 4), phone 1 (Samsung in this case) is better.
Case 2: Nokia is better than Samsung

====================== PHONE ONE INPUT ======================
1).  What brand is your phone? Samsung
2).  What is the condition of your phone? New
3).  Does the phone have Camera?  True
4).  Does the phone have GPS?  False
4).  Does the phone have WiFi?  False
5).  Enter phone price($0-1000) : 900

====================== PHONE TWO INPUT ======================
1).  What brand is your phone? Nokia
2).  What is the condition of your phone? New
3).  Does the phone have Camera?  True
4).  Does the phone have GPS?  False
4).  Does the phone have WiFi?  True
5).  Enter phone price($0-1000) : 900

Nokia is better than Samsung!

In this case we have:

Samsung = 10 (new) + 1 (has camera) + 0 (does not have GPS) + 0 (does not have Wi-Fi) = 11

Nokia = 10 (new) + 1 (has camera) + 0 (does not have GPS) + 1 (does not have Wi-Fi) = 12

Notice that here both cell phones have the same price ($900.00, entered by the user) but they have different estimated values. Based on point 2 defined in the rules cell phone Nokia (phone 2) is better than cell phone Samsung (phone1) because it has a higher value.

Case 3: Both phones are the same

====================== PHONE ONE INPUT ======================
1).  What brand is your phone? Samsung
2).  What is the condition of your phone? New
3).  Does the phone have Camera?  True
4).  Does the phone have GPS?  False
4).  Does the phone have WiFi?  True
5).  Enter phone price($0-1000) : 900

====================== PHONE TWO INPUT ======================
1).  What brand is your phone? Nokia
2).  What is the condition of your phone? New
3).  Does the phone have Camera?  True
4).  Does the phone have GPS?  False
4).  Does the phone have WiFi?  True
5).  Enter phone price($0-1000) : 900

Both phones are the same!

In this third example we have:
Samsung = 10 (new) + 1 (has camera) + 0 (does not have GPS) + 1 (does not have Wi-Fi) = 12

Nokia = 10 (new) + 1 (has camera) + 0 (does not have GPS) + 1 (does not have Wi-Fi) = 12

Both phones have the same estimated value and also the same price (entered by the user). This falls into the point 5 (“Otherwise”) of the rules, which refers to a tie. In this case both phones are the same.

The output of the third case corresponds to just one of the possible cases. For example, it is also possible to have a tie when value of phone 1 is greater than value of phone 2 and price of phone 1 is higher than price of phone 2. Similarly, we have a tie when value of phone 2 is greater than value of phone 1 and price of phone 2 is higher than price of phone 1. Just make sure to treat every case not listed in points 1, 2, 3 and 4 of the rules as indicated at point 5 (The “Otherwise” option).

**Turnin Instructions**

Run putty and login to data.cs.purdue.edu. Turn in your project by typing:

```
data 51 $ cd cs177
data 51 $ turnin -v -c cs177=COMMON –p project1 project1
```

**Grading Rubric**

This project is worth 100 points.

<table>
<thead>
<tr>
<th>TODO</th>
<th>MAX POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 1</td>
<td>60</td>
</tr>
<tr>
<td>Part 2</td>
<td>40</td>
</tr>
</tbody>
</table>