

Python Exercise 2

Wind-Chill Temperature (15 points)

In 2001, the National Weather Service implemented a new wind-chill temperature formula to measure the coldness using temperature and wind speed. The formula is:

$$t_{wc} = 35.74 + 0.6215t_a - 35.75v^{0.16} + 0.4275t_av^{0.16}$$

where:

t_{wc} is the wind chill temperature

t_a is the outside temperature

v is the wind speed

The formula can only be used for temperatures between -58 degrees Fahrenheit and 41 degrees Fahrenheit, as well as wind speeds greater than or equal to 2mph.

Write a program that:

- 1) asks the user to enter a temperature between -58 degrees Fahrenheit and 41 degrees Fahrenheit.
 - Input validation: If the user enters a temperature not in range, use an if statement or while loop to ask them to re-enter the value.
- 2) asks the user to enter a windchill greater than 2mph.
 - Input validation: If the user enters a wind speed not in range, use an if statement or while loop to ask them to re-enter the value.

The program then calculates the wind-chill temperature using the formula above. (You can use the math module for this question if you want.) Format the output to 1 decimal place.

Sample output (including input validation):

```
Enter the temperature in Fahrenheit: -60
Temperature must be between -58F and 41F
Please re-enter the temperature in Fahrenheit: 50
Temperature must be between -58F and 41F
Please re-enter the temperature in Fahrenheit: 35
Enter the wind speed miles per hour: -1
Speed must be greater than or equal to 2
Please re-enter the wind speed miles per hour: 5
The wind chill index is 30.6
```

The Price is Right! (One Bid) (45 points)



Image source: https://priceisright.fandom.com/wiki/One_Bid

“The Price is Right” is a popular game show in the US that focuses on players correctly predicting prices of items for cash and prizes. After players “come on down!” to contestants’ row, the first game that they play is “One Bid” where the players are shown a prize and must determine what the price is. Contestants are not allowed to bid the same amount as another player. The player who is closest to the actual price without going over wins and gets to come up on stage to play for additional prizes. Players who bid the exact prize also win an extra \$500.

Write a program that simulates One Bid. Your code should do the following:

- 1) Generate a random number for the price of the prize. (For this program, set the range of the price as \$1000 - \$5000.)
- 2) Ask four ‘players’ for their bids. All bids should be in whole dollars (no cents).
- 3) Determine if players have over bid. If players have overbid, print a message. (The program would end here.)
- 4) Determine if a player made the exact bid. If so, print a message stating that a player has made an exact bid. (Don’t reveal the price yet, the program will continue.)
- 5) Reveal the price and determine who is the winner. The winner is the person who bids closest to the price (including the exact price) without going over.

You can make the following assumptions:

- Assume that players know that \$1 is the lowest bid they can make (Therefore, your code does not have to check for negative bids).
- Assume that players know they are not allowed to make the same bid. (Therefore, your code does not have to check for other players making the same bid.)

Use the randint method from the random module to generate your prize. Use selection statements to check for overbids from all players, an exact bid and the overall winner.

Sample output is shown below:

Sample output 1 – Everyone overbid. Note that the actual price is not revealed.

Player 1, what is your bid? 3500
Player 2, what is your bid? 3650
Player 3, what is your bid? 5000
Player 4, what is your bid? 3100
Buzz! Aww... everyone has overbid!

Sample output 2 – exact winner. (Extra message printed.)

Player 1, what is your bid? 2500
Player 2, what is your bid? 1100
Player 3, what is your bid? 1350
Player 4, what is your bid? 1560
Ding Ding Ding! One player got it exactly right and gets \$500!
Actual price is \$1100! Player 2, come on up!

Sample output 3 – everyone underbids, closest to prize wins.

Player 1, what is your bid? 800
Player 2, what is your bid? 1150
Player 3, what is your bid? 1976
Player 4, what is your bid? 801
Actual price is \$2814! Player 3, come on up!

Sample output 4 – some players overbid, closest to prize wins.

Player 1, what is your bid? 3500
Player 2, what is your bid? 1500
Player 3, what is your bid? 2750
Player 4, what is your bid? 1
Actual price is \$1541! Player 2, come on up!

Sample output 5 – top the highest value by 1 dollar (a very popular strategy!)

Player 1, what is your bid? 890
Player 2, what is your bid? 756
Player 3, what is your bid? 1350
Player 4, what is your bid? 1351
Actual price is \$2990! Player 4, come on up!

Hint: This code requires careful analysis and design. You may want to ‘plan’ by outlining by hand, how the winner is selected. This can help you with the ‘logic’ of the code. Then, you can build the code by designing it with only two players. Once you have worked out the code for two players, you can then add the third and fourth players.

You can hard code the true price to test the exact win portion of your code. (Just don’t forget to switch it back to random when you are done!) You can also test the code by using the same set of numbers to ensure that each player gets a chance to win.

There are multiple ways to solve this problem, although some strategies are easier than others!

You are allowed to use additional predefined functions or methods to assist you, but they are optional. Just don’t overcomplicate things. If you can’t explain what a function or method is doing or why it works, it’s best to not use it for now. 😊

Here’s a link to see a list python’s pre-built functions. Perhaps some can be useful?

<https://docs.python.org/3/library/functions>