Algorithm Design

LAB 04 : MONOPOLY PROGRAM

Due Saturday at 5:00 PM MST

This week, we will implement in Python a design we created previously.

Program Description

Write a program to inform the user if he or she is able to build a hotel on Pennsylvania Avenue. This program will ask the user several questions and, based on these questions, tell the user whether a hotel can be purchased for Pennsylvania and how much it will cost.

Design

The design for Lab 04 was completed last week. You may use your own flowchart which you developed yourself, or you may use the key provided by your instructor. You can even use some sort of combination of the two if you like.

Prompts

Please use the following prompts:

Prompt: PC

```
What is on Pacific Avenue? (0:nothing, 1:one house, ... 5:a hotel)
```

Prompt: NC

```
What is on North Carolina Avenue? (0:nothing, 1:one house, ... 5:a hotel)
```

Prompt: PA

```
What is on Pennsylvania Avenue? (0:nothing, 1:one house, ... 5:a hotel)
```

Prompt: Cash

```
How much cash do you have to spend?
```

Prompt: Houses

```
How many houses are there to purchase?
```

Prompt: Hotels

```
How many hotels are there to purchase?
```

Prompt: Color Group

```
Do you own all the green properties? (y/n)
```

Make sure your flowchart contains all these prompts and no others.

Output

When finished, the algorithm will display one of the following messages:

Out: Cash

```
You do not have sufficient funds to purchase a hotel at this time.
```

Out: No Houses

```
There are not enough houses available for purchase at this time.
```

Out: No Hotels

```
There are not enough hotels available for purchase at this time.
```

Out: No Properties

```
You cannot purchase a hotel until you own
        all the properties of a given color group.
```

Out: One Hotel

```
You cannot purchase a hotel if the property already has one.
```

Out: Swap NC

```
Swap North Carolina's hotel with Pennsylvania's 4 houses.
```

Out: Swap PC

```
Swap Pacific's hotel with Pennsylvania's 4 houses.
```

Out: Purchase A

```
This will cost $[price].
        Purchase 1 hotel and [number of houses] house(s).
        Put 1 hotel on Pennsylvania and return any houses to the bank.
        Put [number of houses] house(s) on North Carolina.
        Put [number of houses] house(s) on Pacific.
```

Out: Purchase B

```
This will cost $[price].
        Purchase 1 hotel and [number of houses] house(s).
        Put 1 hotel on Pennsylvania and return any houses to the bank.
        Put [number of houses] house(s) on North Carolina.
```

Out: Purchase C

```
This will cost $[price].
        Purchase 1 hotel and [number of houses] house(s).
        Put 1 hotel on Pennsylvania and return any houses to the bank.
        Put [number of houses] house(s) on Pacific.
```

Out: Purchase D

```
This will cost $[price].
        Purchase 1 hotel and [number of houses] house(s).
        Put 1 hotel on Pennsylvania and return any houses to the bank.
```

Hints:

- Try to ask the user the smallest possible number of questions.
- Utilize everything you learned in the week's reading to make the decisions as efficient as possible.

Assignment

To submit this assignment, two things are needed: your source code and a demonstration video.

Source Code

As with Lab 01 and Lab 02, submit your source code as a file attachment in I-Learn. At the top of your program, include a comment answering these five questions:

```
# 1. Name:
#      -your name-
# 2. Assignment Name:
#      Lab 04: Monopoly
```

```
# 3. Assignment Description:
#      -describe what this program is meant to do-
# 4. What was the hardest part? Be as specific as possible.
#      Was it the syntax of Python?
#      Was it an aspect of the problem you are to solve?
#      Was it the instructions or any part of the problem definition?
#      Was it the submission process?
# 5. How long did it take for you to complete the assignment?
#      -total time in hours including reading the assignment and submitting the program-
```

Demonstration Video

Record a short video demonstrating the execution of your program.. The video must be very short. No video longer than two minutes will be accepted. This means you might need to practice once or twice before recording the video to make sure that you demonstrated everything that is necessary. As the video is recorded, mention the test case you are covering.

Your demonstration video must cover the following test cases:

Test Case	Own all green properties	Improvements on Pacific	Improvements on NC	Improvements on PA	Available to buy	Available cash
Does not own enough	no	nothing	nothing	nothing	10 houses, 10 hotels	$1,000
Poor	yes	nothing	nothing	nothing	15 houses, 10 hotels	$100
No houses	yes	nothing	nothing	nothing	10 houses, 10 hotels	$9,000
Swap with Pacific	yes	hotel	4 houses	4 houses	nothing	$0
Swap with NC	yes	4 houses	hotel	4 houses	nothing	$0
Already built	yes	4 houses	4 houses	hotel	10 houses, 10 hotels	$1,000
All at once	yes	nothing	nothing	nothing	12 houses, 3 hotels	$3,000
House and hotel	yes	3 houses	3 houses	3 houses	3 houses, 1 hotel	$5,000

Assessment

Your grade for this activity will be according to the following rubric:

	Exceptional 100%	Good 90%	Acceptable 70%	Developing 50%	Missing 0%
Code Quality 20%	Perfection! The code is extremely easy to understand and very straightforward	Professional and efficient	A few obvious mistakes were made	Readable	Little effort was spent on style. The code looks thrown together or is missing
Decision Quality 30%	Decisions are made as efficiently as possible	One or two decisions were made less than optimally	The user was asked a redundant or unnecessary question	More than one redundant or unnecessary question is asked	Little effort was made on decision quality or no knowledge of the reading was demonstrated
Functionality 40%	All the test cases execute perfectly	Everything works but there are minor cosmetic defects	One test case fails to execute as expected	At least one test case works as expected	Code does not run, is missing, no video, or does not resemble a working solution
Reflection 10%	The reflection component of the assignment completely and concisely describes what went well and what went poorly	It is clear that thought went into the reflection component of the assignment	Each reflection question is addressed, but there is no evidence of introspection	At least one reflection question is answered	There is no reflection