

University of Houston
Department of Computer Science
Homework 4
COSC 1306: Computer Science and Programming
Fall 2016
Sent on: 10/27/2016 6:00 AM
Due: 11/03/2016 11:59 PM

Type – Individual Work

1- (100 points)

For this homework, you need to download the WGC.py file, complete it per instructions, and send it to blackboard with **one output of your code in .txt format**. (Attach completed WGC.py and .txt file in one submission). You have up to three submissions on blackboard and TAs will grade only your last submission.

A farmer is at the east side of the river with a boat, a wolf, a goat and a cabbage. He wants to go to the west side of the river and bring all the three entities with him to the west side. He has a boat with room for only the farmer and one of the entities. So, the farmer has to travel several times between the west and east side of the river to bring all these entities to the east side. When he travels with one entity, he leaves two other entities on their own. If he leaves the goat and the wolf at any side together, the wolf will eat the goat. If he leaves the goat with the cabbage, the goat will eat the cabbage.

Your job is to complete the code in the following functions so that the program can find a solution for this problem. There are two lists for each side of the river. At the beginning, all entities are on the east side and no entity is on the west side (C: Cabbage, F: Farmer, G: Goat, W: Wolf).

```
East=[ 'C' , 'F' , 'G' , 'W' ]  
West=[ ]
```

- The Forbidden list contains all the forbidden combinations that may lead to the goat or the cabbage to be eaten:

```
Forbidden=[ [ 'c' , 'g' , 'w' ] , [ 'c' , 'g' ] , [ 'g' , 'w' ] ]
```

The code contains 4 functions and you need to complete them:

- **GoWest():**
We call this function when the farmer is at the East side. He needs to pick an object and put it in the boat. Remember this action should not cause the goat and the

cabbage, or the wolf and the goat remain at the west side by themselves. Your job is to complete this function considering the mentioned restrictions. Remove the farmer from the East list and add it to the West list. Then, choose an object from the East list randomly and examine if it is safe to move this object with the farmer and leave the two other objects alone. If it is not safe, put the object back to the list and pick another object randomly. Eventually, you will find a safe pick. You are not allowed to hardcode the solution in your program. The only object that can be directly added or deleted from the West or East lists is the farmer. All other three objects should be picked randomly.

- **GoEast():**
Complete this function. We call this function when the farmer is at the West side of the river. He wants to leave to the west side but before that he needs to make sure that by leaving the west side no objects will be eaten! If that is the case, he needs to choose one object randomly and bring it back to the east side.
- **PrintContains(East,West):**
Complete this function so that it prints the objects on the East side and then the objects on the west side. Example:

East:

C

W

West:

G

F

- **Solution():**
So, when should we stop going back and forth between west and East? The Solution() returns True when it reaches the solution (all objects are on the west side) and False otherwise. You need to complete this Function in one line of code.
- The program starts by printing the objects at the east and west side using PrintContains() function. Then using GoWest() and GoEast()the program move the objects till the Solution returns True. Your job is to complete the functions. You are not allowed to change anything in the following code. Adding extra functions is not allowed. This is the main part of the code (you will find it also in the .py file):

```
condition=True
```

```
PrintContains(East,West)
```

```
print('-----')
```

```
while condition:
```

```
    East, West=GoWest(East,West)
```

```

    if not Solution():
        East, West=GoEast(East,West)
    else:
        condition=False

```

- One possible output looks like the following:

East:

C

F

G

W

West:

East:

C

W

West:

G

F

East:

C

W

F

West:

G

East:

C

West:

G

W

F

East:

C

F

G

West:

W

East :
C

West :
W
G
F

East :
C
F
W

West :
G

East :
C

West :
G
W
F

East :
C
F
G

West :
W

East :
C

West :
W
G
F

East :
C
F
G

West :

W

East :

G

West :

W

C

F

East :

G

F

West :

C

W

East :

West :

C

W

G

F

Instructions for sending your homework:

- Python 3.0 is the only language that you should use for this homework. We only accept a .py file for the code and a .txt file for one run of the code.
- Read the instructions of the homework carefully. Do not add anything that was not asked in the instructions. If you are not sure about something, ask TAs by email (allow 24 hours before resending your questions) or attending at their office hours.
- Using objects, dictionaries and all the other materials that has not discussed in the class are not allowed.
- Start early and send your homework on-time. Submitting homework a few minutes before the deadline is not a good idea. Some students found Blackboard unresponsive in the last minutes before the deadline for homework2. To avoid that send your homework at least a few hours before the deadline. If you have other difficulties regarding Blackboard you should ask the UH Blackboard Support:
<http://www.uh.edu/blackboard/support/>. TAs do not have the student view of the blackboard and cannot help you.
- Late submission penalty is 10 points per day after the deadline. Send your late assignment to f.pisheh@gmail.com with the title as:

"1306_Homework4_Lastname_Firstname_PSID_SectionNumber". If you are registered in 11:30-1 class your SectionNumber is 30455 and if you are registered in 1-2:30 class your SectionNumber is 27510.

Good Luck!