1. What wikiHow article did you pick and why?
    a. We picked "How to Open a Coconut". We found that to reach the final goal, there are multiple ways and each way requires various steps. We think this question is complicated enough for us to challenge, and opening a coconut seems like a common scene while surviving in an extreme condition(abandoned island).
2. What portions of the article did you select to translate to PDDL?
    a. We used entire articles to translate to PDDL, and each problem requires different details of the article.
3. Give some examples of the actions, types, and predicates you used in your domain.
    a. Actions:
        i. GO, GET, DROP: we added these actions so that we can make it like a text adventure game
        ii. PIERCE, DRAIN, TURN_ON_OVEN… : we used these to mimic actions introduced in Wikihow.
    b. Types:
        i. appliance - location: Oven is a location, but can be turned on and off
        ii. item, player, direction: these are following the common text adventure settings.
    c. Predicates:
        i. (on ?appliance); this appliance is on
        ii. (at ?obj - object ?loc - location) ; an object is at a location
        iii. (inventory ?player ?item) ; an item is in the player's inventory
        iv. (connected ?loc1 - location ?dir - direction ?loc2 - location) ; location 1 is connected to location 2 in the direction
        v. (wrapped ?item - item); item is wrapped
        vi. (wrapped_with ?item1 - item ?item2 - item); item1 inside item 2
        vii. (burnt ?item); item is burnt
        viii. (pierced ?item); item is pierced
        ix. (smashed ?item); item is smashed
        x. (empty ?item); item is empty
        xi. (peeled ?item); item is peeled
        xii. (test); a test stage for debugging
4. Explain what goal you selected for your problem, and give the initial state and solution that you created.
    a. We selected three goals:
        i. get coconut juice:
            1. initial state: you have everything in this game world, starting with nothing in hand.
            2. solution: get the coconut, knife and glass, pierce a hole on the coconut and pour the juice to the glass.
        ii. get coconut meat without oven
            1. initial state: you have everything in this game world, starting with nothing in hand.
            2. After emptying the coconut, wrap the coconut with a towel and smash it with a mallet, separate it with a knife, and peel with a knife.
        iii. get coconut meat with oven

        1. initial state: you have everything in this game world, starting with nothing in hand.
        2. After emptying the coconut, burn the coconut in the oven, wrap the coconut with a bag and smash it with your hands, separate it with a knife, and peel with a knife.
5. What limitations of PDDL did you encounter that makes it difficult to precisely convert a wikiHow description into PDDL?
    a. I wanted to include the probability of consequences when I applied some actions, but every result is fixed and constant in PDDL.
    b. I had a hard time to track with all preconditions and postconditions while creating PDDL, which makes me use a lot of time to debug
    c. If multiple places and possible actions are included in PDDL, it will take a very long time to find an answer for a problem that requires a long sequence of steps.
6. Could your PDDL be used as an interesting challenge for a text-adventure-style game? If so, how? If not, what would be needed to create an interesting challenge?
    a. Yes. We have already designed our PDDL just like a text adventure game. We have movement between locations, get and drop from inventory, action to create items.
7. Discuss how you might use GPT-3 to automatically or semi-automatically convert a wikiHow article to PDDL?
    a. This is a really hard task. We first need to use GPT-3 to detect actions from each step. And from the first step, we should check inputs(preconditions) and outputs(postconditions) from the text. After extracting all elements, we should construct linkage between steps using information from the text, and fill missing connections by linking words to knowledge graphs, and use triples to link between actions.