Required functions:

1. separatePatternKeywords() string -> list
    a. This function takes the user's input and separates the keywords into a list based on spaces
    b. Calls searchForPattern
2. typeOfCraft() list -> string
    a. This function searches through the list given by separatePatternKeywords() for any word matching "crochet", "cross stitch", "knit", "felting" or anything similar
    b. If no such term exists, move to the next function call
    c. Must check that the item before that word in the list is not "not"
        i. If it is, eliminate that group of crafts from possible search results
    d. Any matching term is added to the user's temporary database
    e. Idea behind this: it narrows down what we have to search through in the database because the database will be separated by craft
    f. Sets craftType to either "", "crochet", "felting", "sewing", "knit", "embroidery", or "cross stitch"
    g. Calls levelOfCraft()
3. levelOfCraft() list -> string
    a. This function searches through the list given by separatePatternKeywords() for any word matching "easy", "beginner", "intermediate", "advanced", "difficult", "hard" or "challenging"
    b. If no such term exists, move to the next function call
    c. Must check that the item before that word in the list is not "not"
        i. If it is, search for the opposite (not easy = intermediate/advanced, not hard = beginner)
    d. Any matching term is added to the user's temporary database
    e. Idea behind this: it is the second level of organization within the database, so it narrows down the search even more
    f. Sets levelOfDifficulty to "", "beginner", "intermediate", or "advanced"
    g. Calls typeOfProduct()
4. typeOfProduct() list -> string
    a. This function searches through the list given by separatePatternKeywords() for any word matching a common type of pattern (clothing, homeware, etc.)
    b. If no such term exists, move to the next function call

    c. Must check that the item before that word in the list is not "not"
        i. If it is, eliminate that group of crafts from possible search results
    d. Any matching term is added to the user's temporary database
    e. Idea behind this: it is the third level of organization within the database, so the search is narrowed down more
    f. Sets productType to either "", "clothing", "accessory", "homeware", or "stuffed animal"
    g. Calls narrowSearch()

5. narrowPatternSearch() input from form -> null
    a. This function takes the inputs from the form that the user could optionally fill out and narrow down the search even further
    b. If no such input exists, move to the next function call
    c. Checks for difficulty, type of craft, type of product
    d. If one of the string variables is empty from the previous three functions but there is an input from the form, replace the "" of the function with the specifications given
    e. Call compilePatternResults()
6. compilePatternResults strings -> list
    a. Puts all strings into a list in this exact format: (craftType, levelOfDifficulty, productType)
    b. Calls pullFromDatabase()
7. pullFromDatabase() list -> results on screen
    a. Based on the list of terms in the user's temporary database, this function navigates through the database of terms based on the level of organization to return the closest search result possible
    b. If the original input from the user exactly matches something in the database, pull that and then the next 10 results
    c. Otherwise, pull patterns from the exact level of organization that was narrowed down to
    d. On the first page, pull the closest 10 results

Required Variables:

1. separatedPatternSearch : list
    a. Created in the separatePatternKeywords() function
2. originalPatternSearch : string
    a. Unaltered input from the user in the search bar
3. levelOfDifficulty : string

    a. Is either “” or “beginner”, “intermediate”, or “advanced”

4. craftType : string
    a. Is either “”, “crochet”, “felting”, “sewing”, “knit”, “embroidery”, or “cross stitch”
5. productType : string
    a. Is either “”, “clothing”, “accessory”, “homeware”, or “stuffed animal”
6. finalPatternSearch : list
    a. List created in compilePatternResults()