**HOMEWORK 3**

(handout for students)

**42 marks in total**

This homework aims to test your problem-solving skills and challenge you to apply coding concepts from the last few weeks. The questions in this homework may be similar to what you’d face in a coding interview and in certain sections of the Final Assessment.

For some solutions you will be asked to judge the efficiency, a concept which will be taught in the first session of next week. You can either do external research beforehand, or wait until the topic is taught to complete.

|  |  |
| --- | --- |
| **QUESTION 1: Jumping The Queue** | **15 marks** |

**A] Write a program that takes in an input file and prints out a list with the final order of who’s in the queue.**

*You are expected to use the provided input file hw3\_q1.txt to develop and test your solution.*

The first word on each line will either be “JUMP” or “JOIN” and the second word the name of the person, the word “JUMP” means the person in question has gone to the front of the queue, if the word is “JOIN” it means they have joined the back of the queue.

You are expected to identify and use a container from the collections module for your solution.

**from** collections **import** deque  
  
  
**with** open(**'hw3\_q1.txt'**, **'r'**) **as** file:  
 lines = file.readlines()  
  
  
queue = deque()  
  
**for** line **in** lines:  
 command, name = line.strip().split()  
 **if** command == **'JUMP'**:  
 queue.appendleft(name)  
 **elif** command == **'JOIN'**:  
 queue.append(name)  
  
print(**"Final order of the queue:"**)  
**for** person **in** queue:  
 print(person)  
*#the time complexity is O(n) here, n is the number of lines in the input file.the space complexity is O(n) where n is the number of people in the quese.  
#Assumption: the input file has input in correct format and accessible. it is not a large file.*

**B] What is the time and space complexity of your solution? If you are making any assumptions, list them.**

Add your answer to this as a comment above or below your solution to Part A.

|  |  |
| --- | --- |
| **QUESTION 2: Number In Sequence** | **12 marks** |

**A] Write a function that returns the Nth number in the following sequence**

8, 15, 22, 29, 36, …

For example:

* num\_in\_seq(1) = 8
* num\_in\_seq(5) = 36
* num\_in\_seq(10) = 71

**Non-recursive solution: 6 marks**

*#non-recursive solution:***def** num\_in\_seq(n):  
 **if** n <= **0**:  
 **return None** first\_number = **8** difference = **7** nth\_number = first\_number + (n - **1**) \* difference  
 **return** nth\_number  
  
n = int(input(**"Enter the value of N: "**))  
result = num\_in\_seq(n)  
print(**f"The** {n}**th number in the sequence is:** {result}**"**)

**Recursive solution: 12 marks**

**def** num\_in\_seq\_recursive(n):  
 **if** n == **1**:  
 **return 8  
 else**:  
 **return** num\_in\_seq\_recursive(n - **1**) + **7**n = int(input(**"Enter the value of N: "**))  
result = num\_in\_seq\_recursive(n)  
print(**f"The** {n}**th number in the sequence is:** {result}**"**)

|  |  |
| --- | --- |
| **QUESTION 3: Hyperlink History** | **15 marks** |

**A] Simulate clicking around the CFG Website. Keep track of the URL changes and print the current URL after each move.**

* You will need to display the options for each link, and include an option for ‘Back’ if not on the Base URL.
* You do not need to worry about error handling
* You are recommended to keep the simulation going within a **while True** loop so the logic keeps looping until an exit is forced.

*You* ***only*** *need to consider the following URLs for your solution:*

1. Base URL: <https://codefirstgirls.com/>
2. Category URLs: [/courses](https://codefirstgirls.com/courses) ,[/opportunities/](https://codefirstgirls.com/opportunities/)
3. Sub-category URLs: /[courses/cfgdegree/](https://codefirstgirls.com/courses/cfgdegree/) , [/opportunities/ambassadors/](https://codefirstgirls.com/opportunities/ambassadors/)

Example run of the program:

|  |
| --- |
| You are currently on the URL <https://codefirstgirls.com/>  Where are you clicking?  Options: Courses, Opportunities  Courses  You are currently on the URL https://codefirstgirls.com/courses  Where are you clicking?  Options: CFGDegree, Back  CFGDegree  You are currently on the URL https://codefirstgirls.com/courses/cfgdegree  Where are you clicking?  Options: Back  Back  ou are currently on the URL https://codefirstgirls.com/courses  Where are you clicking?  OptiYons: CFGDegree, Back  Back  You are currently on the URL https://codefirstgirls.com/  Where are you clicking?  Options: Courses, Opportunities  Opportunities  You are currently on the URL https://codefirstgirls.com/opportunities  … |
| base\_url = **"https://codefirstgirls.com/"** category\_urls = [**"/courses"**, **"/opportunities"**] subcategory\_urls = [**"/courses/cfgdegree"**, **"/opportunities/ambassadors"**]  current\_url = base\_url  **while True**:  print(**"You are currently on the URL"**, current\_url)  print(**"Where are you clicking?"**)   **if** current\_url == base\_url:  print(**"Options: Courses, Opportunities"**)  **elif** current\_url == base\_url + **"courses"**:  print(**"Options: CFGDegree, Back"**)  **elif** current\_url == base\_url + **"opportunities"**:  print(**"Options: Ambassadors, Back"**)  **elif** current\_url == base\_url + **"courses/cfgdegree"**:  print(**"Options: Back"**)  **elif** current\_url == base\_url + **"opportunities/ambassadors"**:  print(**"Options: Back"**)   choice = input().strip().lower()   **if** current\_url == base\_url:  **if** choice == **"courses"**:  current\_url = base\_url + **"courses"  elif** choice == **"opportunities"**:  current\_url = base\_url + **"opportunities"  elif** current\_url == base\_url + **"courses"**:  **if** choice == **"cfgdegree"**:  current\_url = base\_url + **"courses/cfgdegree"  elif** choice == **"back"**:  current\_url = base\_url  **elif** current\_url == base\_url + **"opportunities"**:  **if** choice == **"ambassadors"**:  current\_url = base\_url + **"opportunities/ambassadors"  elif** choice == **"back"**:  current\_url = base\_url  **elif** current\_url == base\_url + **"courses/cfgdegree" or** current\_url == base\_url + **"opportunities/ambassadors"**:  **if** choice == **"back"**:  current\_url = base\_url + **"courses" if** current\_url.startswith(  base\_url + **"courses"**) **else** base\_url + **"opportunities"** print()   *#The time and space complexity of this solution is O(1). because it is not a recursive function or no loop is present. the number of urls are fixed and the options are limited. the execution time remains same. #Assumption: all the urls and options are not changing and limited. the input matches the options provided. it doesnot have any extensive error handling and input validation. so the time and space complexity can be thought of as constant.as the input size does not vary with the users. the inputs are predetermined.* |

**B] What is the time and space complexity of your solution? If you are making any assumptions, list them.**

Add your answer to this as a comment above or below your solution to Part A

|  |
| --- |
| SUBMISSION CRITERIA:   * This homework is to be submitted solely via GitHub. * Your pull request needs to contain all the Python solutions (either in one file or 3 separate ones) * The answers for complexity should be comments given above or below your solution - these need to be clear and annotated well. |