Python Programming

Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Homework 1: Find most frequent number

- Read a line of N integers. Each integer is -500 <= value <= 270
- Find the value that repeated the most number of times.
  - If there are many solutions: find the **smallest** value
- Input ⇒ output
  - -1 2 -1 3 -1 5 5 ⇒ Value -1 repeated 3

Homework 2: Digits frequency

- Read a line of N integers.
- Compute the digits [0 to 8] frequency of all the N numbers
  - Input 78 307 [compute digits frequency of 7 8 3 0 7]
  - Output:
  - 0 1
  - 1 0 [digit 1 never appeared]
  - 2 0
  - 3 1
  - 4 0
  - 5 0
  - 6 0
  - 7 2 [digit 7 appeared twice]
  - 8 1
  - 9 0

Recall Definitions for sequences

- Let’s say we have string aaabcdefgg
- **Prefix**: Any string starts from the first character (n prefixes)
  - a, aa, aaa, aaab, aaabc, ….. aaabcdefgg
- **Suffix**: Any string sends at the last character (n suffixes)
  - g, gg, fgg, efgg, …. Aaabcdefgg
- **Substring**: Starts wherever and end wherever, but **consecutive**
  - E.g. of length 3: aaa, aab, abc, cde, def, efg, fgg. Same as *subarray*.
- **Sub-sequence**: Not **consecutive** but must be in **order**
  - In order: Next letter must has bigger index
  - adef, bgg, aeg, cdgg
    - aeg indices: 0 5 8
  - But not: gga, ed, aca

Homework 3: Is subsequence

- Read a line of N integers. Let's call it list1
- Then Read a line of M integers. Let's call it list2
- Print YES if list2 is a subsequence of list1. Otherwise print NO
- Input ⇒ Output
  - [1 2 3 4] [1 4] ⇒ True
  - [1 2 3 4] [4 1] ⇒ Fase (items exist but NOT in order)
  - [10 -10 20 25 2 7 2 3] [-10 2 2 3] ⇒ True
  - [10 -10 20 25 2 7 2 3] [-10 2 2 2 3] ⇒ False
- Can you do it in a single loop?

Homework 4: Recamán's sequence

- The first terms of this sequence are 0, 1, 3, 6, 2, **7**, …
  - So last term **value** is 7 and its **index** is 5 (zero based)
  - The next value is either:
    - **LastValue - LastIndex - 1** if the following 2 conditions are satisfied:
      - value > 0 and It did not appear before
      - E.g. 7 (last value) - last index (5) - 1 = 7-5-1 = 1 (> 0 but already exists)
    - Or **LastValue + LastIndex + 1** = 7+5+1 = 13
- Read integer zero-based index ([1, 200]) and print the value of this index
  - E.g. (6 ⇒ 13), (9 ⇒ 21), (17 ⇒ 25)
- Don't use nested loops
- The series is: 0, 1, 3, 6, 2, 7, **13**, 20, 12, **21**, 11, 22, 10, 23, 9, 24, 8, **25**, 43

Homework 5: Remove evens inplace

- Read a line of N integers.
- Implement function: `def remove_evens_inplace(lst):`
  - It finds all the even numbers and remove them **in place**
  - Try to do it without creating new memory
- Input ⇒ Output
- 1 2 3 4 5 6 ⇒ 1 3 5
- -6 6 ⇒ Empty output
- Empty input ⇒ Empty output

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."