CPS 2231 2025 Spring
Lab2: Review of CPS 1231

Instructor: Dr. Y. Tiffany Tang[1] and Dr. Pinata Winoto[2]
Songlin Shang[3], Cong Ye[4] and Zike Deng[5]

Instruction:

Self-check: In all labs in the future, you can download `check.py` in canvas, which will help you check the output format in labs, to make sure you don't have compile error or logic error before you submit your file. If you didn't finish installing the python environment/JDK installation, you can download the file from canvas named as tutorial.pdf. Please read and follow the tutorial, which teaches you how to install a python environment, for checking your code result. Also, the instructions of all questions also teach you how to use `check.py` to check your code.

Plagiarism: We encourage discussing all labs with your classmate, but **NEVER** copy other's code directly. We will use GradeScope[@] platform, to check the similarity probability, 90% similarity reported by GradeScope will be reviewed by TAs, and will report to your instructor and receive some penalty based on the event.

AI-Generate code issue: We encourage you to use AI tools, to help you compile all labs, **BUT NEVER** directly **copy and paste**. To detect that, we will use different tool to detect AI possibility include:

ChatGPT Code Detector:https://chatgpt.com/c/67bdddec-c728-8010-b1b6-abf3e56b3136

GPTSniffer: https://github.com/MDEGroup/GPTSniffer

Penalty will be given for high AI probability.

Grading: Each lab has the same points, and each test case has the same points. In check.py, you might check your code with given test cases in labs to avoid spelling errors and compile errors. We still have invisible test cases, which test for grading. This might test you to consider all possible positions. All test cases have the same points.

[1] Tiffany Ya Tang, Associate Professor, Department of Computer Science, Wenzhou-Kean University; Email: yatang@kean.edu
[2] Pinata Winoto, Assistant Professor, Department of Computer Science, Wenzhou-Kean University; Email: pwinoto@kean.edu
[3] Songlin Shang, Department of Computer Science, Wenzhou-Kean University; Email: shangs@kean.edu
[4] Cong Ye, Department of Computer Science, Wenzhou-Kean University; Email: yecon@kean.edu
[5] Zike Deng, Department of Computer Science, School of Computing and Data Science, University of Hong Kong; Email: baylordeng@connect.hku.hk

Get Started

Please download Lab2_1.java, Lab2_2.java, Lab2_3.java on canvas.

Before you start the lab, you can create a project on eclipse called Lab2. Also, we suggest you download the check.py in canvas and move it under the same folder with all labs.

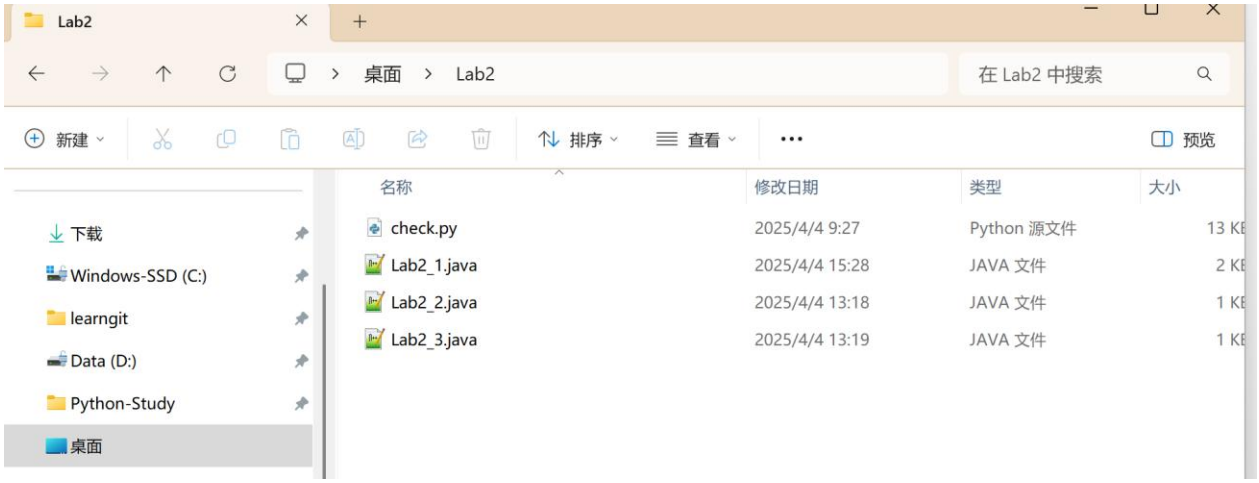
This would be like:

check.py	2025/4/4 9:27	Python 源文件	13
Lab2_1.java	2025/4/4 15:28	JAVA 文件	2
Lab2_2.java	2025/4/4 13:18	JAVA 文件	1
Lab2_3.java	2025/4/4 13:19	JAVA 文件	1

Before you start coding, please direct you terminal to same folder using following code:

`cd path (path is the way to the folder of check.py)`

here is an example:



Now you can start coding after all these preparations.

Lab2_1
Write a method that uses loops to compute Fibonacci numbers and print the first Fibonacci numbers.

> **N.B.**: **Please replace Li Hua with your Chinese name in Pinyin.**
> **You are not allowed to use knowledge you haven't learned in CPS 1231 and 2231.**

Example 1

Input: 12

Output: After Li Hua's contribution, Fibonacci Series till 12 terms: 1 1 2 3 5 8 13 21 34 55 89 144

Example 2

Input: 5

Output: After Li Hua's contribution, Fibonacci Series till 5 terms: 1 1 2 3 5

Example 3

Input: n

Output: Invalid Input.

Example 4

Input: n

Output: Invalid Input.

Example 5

Input: 0

Output: Invalid Input.

Self-Check:

When you finish coding, you can run following code:

```
python check.py --lab Lab2_1
```

If everything good, you might see following content:

```
********* Compiling Java file: Lab2_1.java *********

☑ Compilation succeeded!

Running test case: Valid input 12
```

********* Running Java program: Lab2_1 *********

☑ Execution succeeded!

☑ Output is correct: After Li Hua's contribution, Fibonacci Series till 12 terms: 1 1 2 3 5 8 13 21 34 55 89 144

Running test case: Valid input 5

********* Running Java program: Lab2_1 *********

☑ Execution succeeded!

☑ Output is correct: After Li Hua's contribution, Fibonacci Series till 5 terms: 1 1 2 3 5

Running test case: Invalid input a

********* Running Java program: Lab2_1 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

Running test case: Invalid input -1

********* Running Java program: Lab2_1 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

Running test case: Invalid input 0

********* Running Java program: Lab2_1 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

☑ All checks passed for the specified lab!

Lab 2_2
Write a method that use 2D-array in creating a Multiplicative Table

N.B.: **Please replace Li Hua with your Chinese name in Pinyin.**
You are not allowed to use knowledge you haven't learned in CPS 1231 and 2231.

Example 1
Input: 5 10
Output: Li Hua's multiplicative table is:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50

Example 2
Input: 3 5
Output: Li Hua's multiplicative table is:

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15

Example 3
Input: a b
Output: Invalid Input.

Example 4
Input: -1 10
Output: Invalid Input.

Example 5
Input: 0 0
Output: Invalid Input.

Self-Check:

When you finish coding, you can run following code: `python check.py --lab Lab2_2`

If everything good, you might see following content:

```
********* Compiling Java file: Lab2_2.java *********

☑ Compilation succeeded!

Running test case: Valid input 5 10

********* Running Java program: Lab2_2 *********
```

☑ Execution succeeded!

☑ Output is correct: Li Hua's multiplicative table is:
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50

Running test case: Valid input 3 5

********* Running Java program: Lab2_2 *********

☑ Execution succeeded!

☑ Output is correct: Li Hua's multiplicative table is:
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15

Running test case: Invalid input a b

********* Running Java program: Lab2_2 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

Running test case: Invalid input -1 10

********* Running Java program: Lab2_2 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

Running test case: Invalid input 0 0

********* Running Java program: Lab2_2 *********

☑ Execution succeeded!

☑ Output is correct: Invalid Input.

☑ All checks passed for the specified lab!

Lab 2_3 Challenge Question (Optional)

Find the Starting Point of a Cycle in an Array Jump Path

Background Story:

In a mysterious numerical maze, an array records the paths of the maze. Each element in the array represents the next index to jump to (starting from 0). The maze designer might have left a cycle, or they might have designed an exit (jumping beyond the array's bounds). Your task is to determine if there is a cycle in the path. If a cycle exists, return the starting index of the cycle; if not, return -1.

Problem Description:

Given an integer array nums, where each element is a non-negative integer representing the next index to jump to from the current position, and the array length is n, if nums[i] >= n, it means the jump goes beyond the array bounds, and the path terminates. If a cycle is formed through jumping (i.e., returning to a previously visited index), return the starting index of the cycle; otherwise, return -1.

Requirements:

Return value:

- If a cycle exists, return the index of the cycle's starting point (the first index visited repeatedly).
- If no cycle exists, return -1.

N.B.:
You are not allowed to use knowledge you haven't learned in CPS 1231 and 2231.

Example 1
Input: nums = [1, 3, 4, 2, 5, 3]
Output: 3
Explanation:
Starting from index 0: 0 → 1→ 3→ 2 → 4→ 5 → 3(cycles back to 3)
The cycle is [3, 2, 4, 5], with the starting point at index 3 .

Example 2
Input: nums = [1, 2, 3, 4]
Output: -1
Explanation:
Starting from index 0: 0 -> 1 -> 2 -> 3 -> 4 (exceeds array length 4, terminates)
No cycle, return -1.

Example 3

Input: nums = [1, 2, 0]
Output: 0
Explanation:
Starting from index 0: 0 -> 1 -> 2 -> 0 (cycles back to 0)
The cycle is [0, 1, 2], with the starting point at index 0.

Example 4
Input: nums = [3, 1, 2, 0]
Output: 0
Explanation:
Starting from index 0: 0 -> 3 -> 0 (cycles back to 0)
The cycle is [0, 3], with the starting point at index 0.

Hints:

- **Introduction to the Fast-Slow Pointer Method**: Use two pointers—"slow" and "fast". The slow pointer moves one step at a time (slow = nums[slow]), while the fast pointer moves two steps at a time (fast = nums[nums[fast]]). If they meet, a cycle exists; if a jump exceeds the array bounds, there is no cycle.
- Check if a jump exceeds the array length n to avoid out-of-bounds errors.
- If a cycle exists, adjust the pointers to find the starting point; if no cycle exists, return -1.

Self-Check:

When you finish coding, you can run following code: `python check.py --lab Lab2_3`
If everything good, you might see following content:

```
(base) baylordeng@BaylordeMacBook-Pro Lab2 % python check.py --lab Lab2_3

********* Compiling Java file: Lab2_3.java *********

☑ Compilation succeeded!

Running test case: Example 1: cycle starts at 3

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 3

Running test case: Example 2: no cycle

********* Running Java program: Lab2_3 *********
```

☑ Execution succeeded!

☑ Output is correct: -1

Running test case: Example 3: cycle starts at 0

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 0

Running test case: Example 4: cycle starts at 0

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 0

Running test case: Single element cycle

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 0

Running test case: Single element jumps out

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: -1

Running test case: Two elements forming a cycle

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 0

Running test case: Path jumps out of array

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: -1

Running test case: Cycle 0→2→1→0

********* Running Java program: Lab2_3 *********

☑ Execution succeeded!

☑ Output is correct: 0

☑ All checks passed for the specified lab!

Last Step

Congratulations! You have finished all Labs in Lab2.

Now please run the following code in the end:

```
python check.py --uid 123456
```

(replace 123456 as you student number)

It will check all labs again and zip a file. Since Lab 2_3 is an optional lab, it will ask you to answer whether you have finish the lab, and you might see following content:

Have you completed Lab2_3? (y/n):

Input "`y`",then press "`enter`" if you finish Lab2_3, **otherwise press n.**

If everythings good, you might see following information:

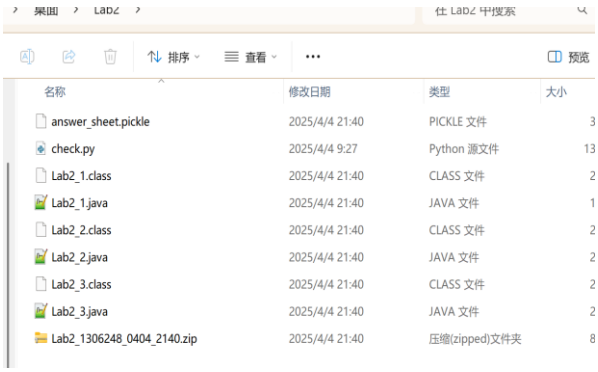
Your Student ID is 123456. Is this correct? (y/n):

Input "`y`",then press "`enter`", you will see following information.

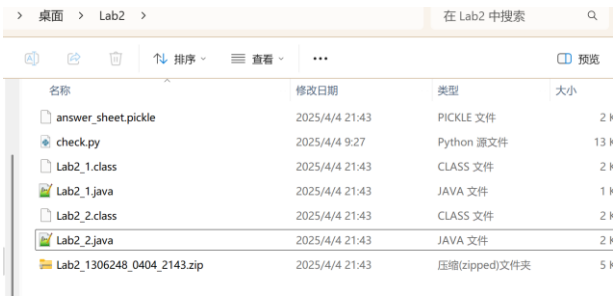
********* Zipping Files *********
Zipping Lab2_1.java... Done.
Zipping Lab2_2.java... Done.
Zipping answer_sheet.pickle... Done.
Zipping Lab2_3.java... Done.

Successfully created submission package: Lab2_123456_0404_2051.zip

You will see there is a new file under the folder like:



or



Submit this .zip file to canvas!