Welcome !

2024 Fall CS101 Introduction to Programming

KAIST School of Computing

Week 6

Sequence
(Lists, Strings, and Tuples)

Week 6
Today’s Tasks

Tasks for Today!

- Tasks
  - Grading
  - Drawing integers
  - Memento game
  - Hidden task
- Tip: How to get the information

Task 1 | Grading

In a list of tuple named theory_point_list, each tuple element has the theory point and name of one student as follows:

theory_point_list = [(27, ‘Russell Shharp’), (77, ‘Egbert Booth’) …]

Implement following two functions using **List Slicing**

1. list_of_A0(theory_point_list)

- Students with the top 8th to 15th of thoery points will receive an A0.

- Return a list of names of students who received an A0.

- Students' names should be in alphabetical order.

Task 1 | Grading

2. average_Aplus(theory_point_list)

- Students with the top 7 of theory points will receive an A+

- Return the average score of student who received an A+.

Task 2 | Drawing Integers

1. **drawing_integers**(lb, ub, trials)
   - Goal: Make a list of random integers
   - Parameters
     - Range of the integers: lb <= the integers <= ub
     - Length of the integers: trials
   - Return: A list of Integers.

     Ex) [1, 4, 3, 5, 2]
   - Hint: Use randint() function in the random module
2. **average_integers**(num_list)
   - Goal: Compute the average of the integer sequence in the list
   - Parameter: a list which is returned from drawing_integers()
   - Return: the average value of the list

Task 2 | Drawing Integers

3. **count_integers**(num_list)
   - Goal: Count the frequency of the integers in the list
   - Parameter: a list which is returned from drawing_integers
   - Return: A list of tuples that are the integer and its frequency

     Ex) [(1, 2), (2, 3), (3, 0), (4, 2)] for [4,1,2,2,4,2,1]

Task 2 | Drawing Integers - Example

```
list1 = drawing_integers(1, 6, 20)
print(list1)
print(average_integers(list1))
print(count_integers(list1))
print()
list2 = drawing_integers(5, 12, 15)
print(list2)
print(average_integers(list2))
print(count_integers(list2))
```

Code

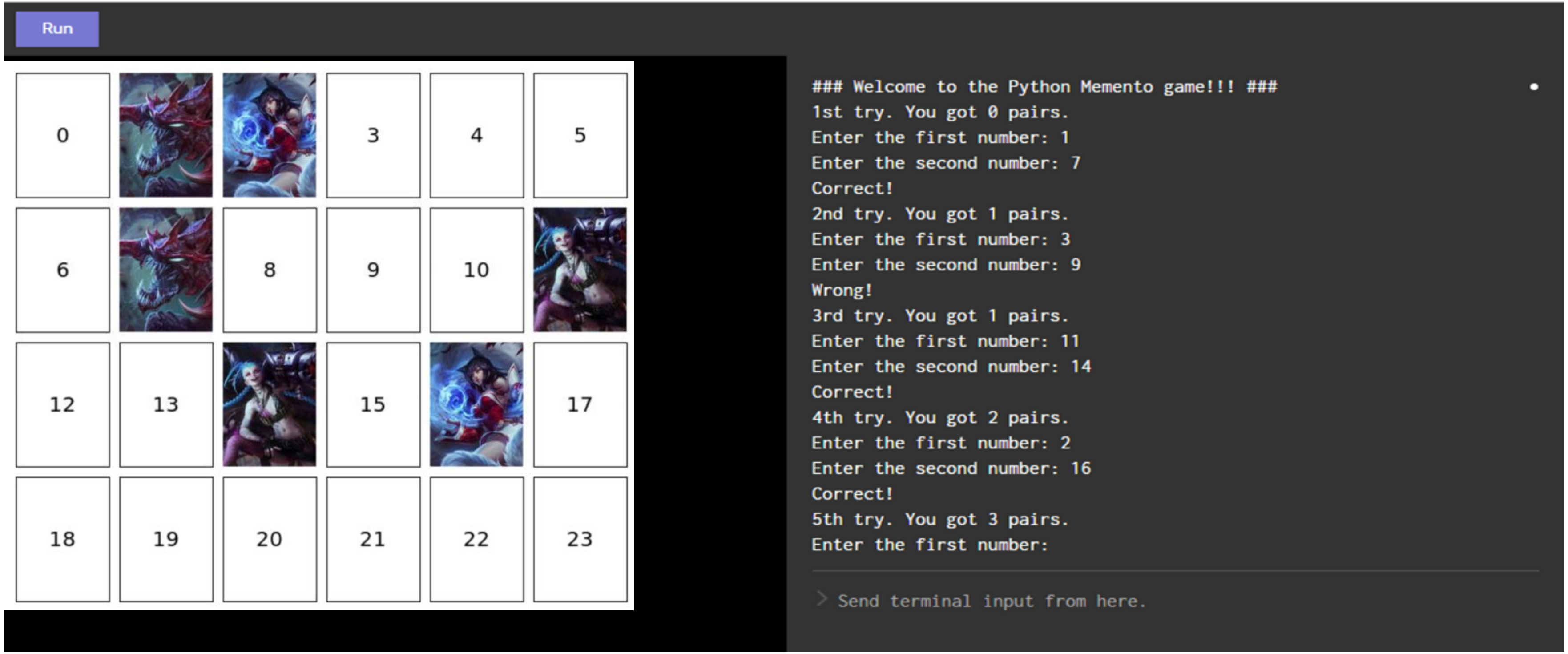
```
[6, 1, 6, 4, 5, 1, 4, 1, 1, 5, 3, 2, 3, 4, 5, 5, 5, 6, 6, 1]
3.7
[(1, 5), (2, 1), (3, 2), (4, 3), (5, 5), (6, 4)]

[11, 11, 11, 8, 10, 9, 6, 5, 5, 10, 12, 6, 9, 9, 11]
8.86666666667
[(5, 2), (6, 2), (7, 0), (8, 1), (9, 3), (10, 2), (11, 4), (12, 1)]
```

Result

Task 3 | Memento Game

Today we'll implement a Memory test game, "Memento"

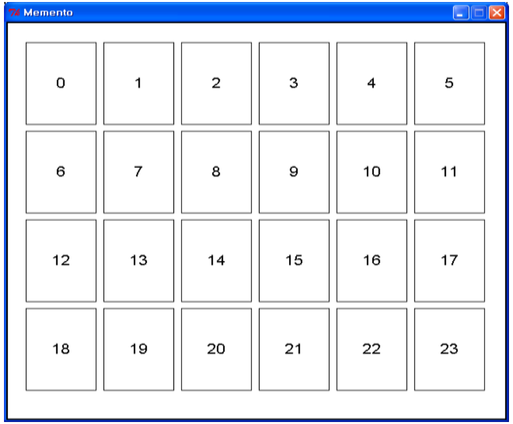


```
### Welcome to the Python Memento game!!! ###
1st try. You got 0 pairs.
Enter the first number: 1
Enter the second number: 7
Correct!
2nd try. You got 1 pairs.
Enter the first number: 3
Enter the second number: 9
Wrong!
3rd try. You got 1 pairs.
Enter the first number: 11
Enter the second number: 14
Correct!
4th try. You got 2 pairs.
Enter the first number: 2
Enter the second number: 16
Correct!
5th try. You got 3 pairs.
Enter the first number:
```

Task 3 | The Flow of Memento Game

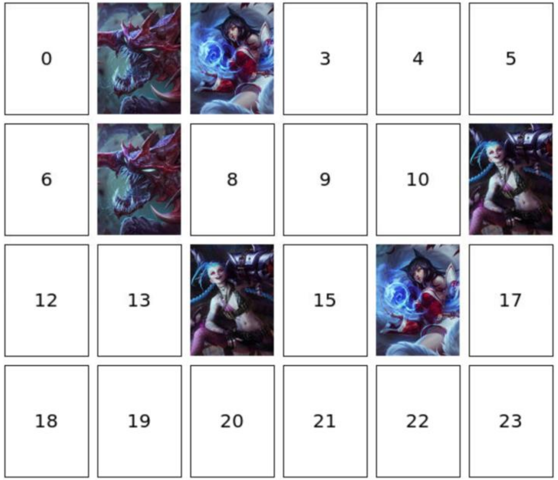
- There are 24 number pads
- First, show all pictures for a while then flip them over
- The user have to input 2 card numbers
- The screen shows two pictures the user indicated
- If the card numbers indicate the same picture
  - Show “Correct!” message
  - Add 2 numbers to correct number list
  - Show all the picture pairs which you have founded
- If not
  - Show “Wrong……” message
  - Return to the screens before
- The game finishes when all pairs are found

```
>>> [evaluate memento.py]
### Welcome to the Python
Memento game!!! ###
1st try. You got 0 pairs.
Input the first card number :
```

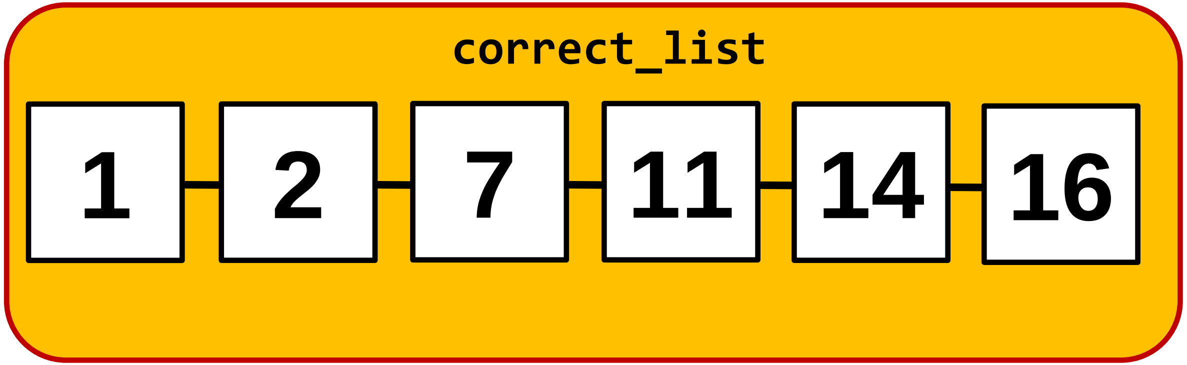


0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23

Task 3 | Three Lists



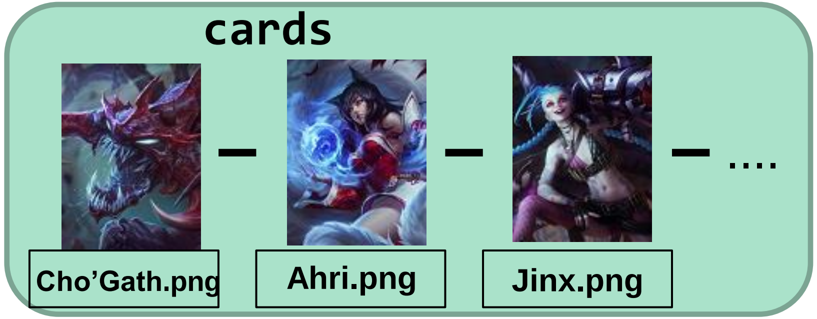
A list for managing corrected numbers



correct_list

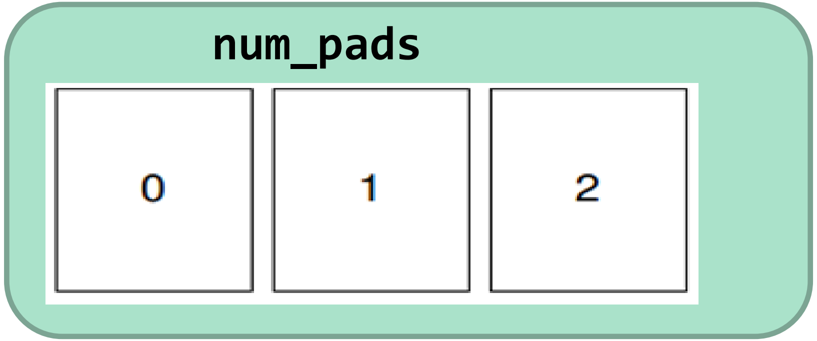
1 – 2 – 7 – 11 – 14 – 16

Two lists for visualization



cards

Cho'Gath.png – Ahri.png – Jinx.png – ....



num_pads

0 1 2

Task 3-1 | Implementation

1. **is_valid**(num1, num2)
    - Check if the two numbers
        - exist in the current correct list
        - are same number
        - are valid numbers in the given range.
    - Return Boolean value according to the result
2. **check**(num1, num2)
    - At first, visualize the screen including the two cards (num1th card and num2th card)
    - If two pictures of the two cards are same,
      put the two numbers into the correct list
    - If not, re-visualize the original screen
    - Return Boolean value according to the result

Task 3-2 | Implementation

1. Shuffle the “cards” list
   - Using “random” module
2. Modify the condition of main while-loop
   - Your program should be terminated if you find all pairs
3. Print the number of tries using the ordinal number
4. Update the number of trials

```
>>> [evaluate memento.py]
### Welcome to the Python Memento game!!! ###
1st try. You got 0 pairs.
2nd try. You got 0 pairs.
3rd try. You got 0 pairs.
….
```

Hidden Task

- Today we have a hidden task which is not revealed until the lab starts
- The hidden tasks are different for each section
- Complete the hidden task corresponding to your section

Rock and Roll