SI100B Final Exam (Python Part) Test Book

Instructor: Yue Qiu, Haipeng Zhang
June 23, 2020

PRINT your FULL CHINESE NAME: __________________________

PRINT your email ID: ___________________@shanghaitech.edu.cn

PRINT your student ID: __________________________

INSTRCTIONS:

1. Full score of the Final exam is 150 points with 4-point bonus.
2. You have 150 minutes (18:00~20:30) to complete this exam. You could submit your answer book in advance before 20:15.
3. Your exam will **not** be graded if you fail to write down the Academic Integrity Protocols and sign your full Chinese name in the required places on the first page of the Answer Book.
4. Since the Final Exam will be detached and scanned into Gradescope to grade online, you are required to write down your email address prefix on **every** page in the required places on the page headers. This task is worth 2 points.
5. This exam is closed book. Bring only stationery, some tissues, some drinks and some snacks (if you want) to your examination seat.
6. Please write all your answers into the Answer Book. The Test Book and scratch paper will not be collected and graded after the exam.
7. **English is the only accepted language for the Final.** All answers using languages other than English will not be graded.

STOP! DO NOT turn this page until the instructor tells you to do so!

Section 0: Tell Me Who You Are

(2 pts)

0. Please write down the Academic Integrity Protocols correctly and sign your full Chinese name correctly in the required place on the cover of the Answer Book. We will not grade your answers if you fail to do so.

1. (2 pts) Please identify yourself by:

- Writing down your **FULL CHINESE NAME**, your **email ID** and **student ID** in the **required** place on the **cover** of the **Answer Book**.
- Writing down your **email ID** on **every page** in the **required** places on the **page headers**.

You will get the 2 points only when complete **both** tasks successfully. Otherwise, you will lose them.

If we fail to identify who you are, your exam will be graded as 0 point.

Section 1: Multiple Choices with Only One Correct Answer

(2.5 pts * 16 = 40 pts) *Each question has **only one correct answer**. **Choose the correct answer**. All the following programs are executed in the interactive `Python3` interpreter.*

2. What gets printed if the following program is executed by an interactive Python 3 interpreter?

```
for i in [i for i in range(3)]:
    print(i, end = ' ')
```

A. It will cause an error.

B. `0 1 2`

C. `1 2 3`

D. `[1] [2] [3]`

3. What gets printed if the following program is executed by an interactive Python 3 interpreter?

```
a = 2**2**(3**0*2+4.5//2/2)
print(a)
```

A. `256.0`

B. `4.0`

C. `64.0`

D. `16.0`

4. What gets printed if the following program is executed by an interactive Python 3 interpreter?

```python
a = [1,2]
b = [3,4]
print(a+b, end = ' ')
a = (1,2)
b = (3,4)
print(a+b)
```

A. `[1,2,3,4] (1,2,3,4)`

B. `[1,2,3,4] (4,6)`

C. `[[1,2], [3,4]] (4,6)`

D. `[[1,2], [3,4]] ((1,2), (3,4))`

E. It will cause an error.

5. Suppose `myfile.txt` is in the same directory with the script file storing the codes below. The content inside `myfile.txt` is:

```
abcd
```

What gets printed if the following program is executed by an interactive Python 3 interpreter?

```python
import sys

try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
    raise
else:
    print("Success, int =",i)
finally:
    print("over")
```

A. Prints `OS error:`, some information of `OSError` and `over`.

B. Prints `Could not convert data to an integer.`, `over`.

C. Prints `Unexpected error:` and some information, `over` and re-raise the error.

D. Prints `Success, int = abcd` and `over`.

6. Suppose `data.txt` is in the same directory with the script file storing the codes below. Its content is:

```
We_all_love\nShanghaitech
```

What gets printed if the following program is executed by an interactive Python 3 interpreter?

```python
with open('data.txt', 'r', encoding='utf-8') as fl :
    fl.seek(3)
    lines = fl.readlines()
    print(lines)
print(fl.closed)
```

A.

```
['_all_love\n', 'Shanghaitech']
True
```

B.

```
['_all_love', '\nShanghaitech']
True
```

C.

```
['all_love', '\nShanghaitech']
False
```

D.

```
['all_love\n', 'Shanghaitech']
True
```

7. What gets printed if the following program is executed by an interactive Python 3 interpreter?

```python
class A:
    def __init__(self, x = 2, y = 0 ,z = 2 , s = 0):
        self.x = x
        self.y = y
        self.z = z
        self.s = s

    def __eq__(self, num ):
        return self.x * self.z * (self.y + self.s) == num.x * num.z * (num.y + num.s)

def main():
    a = A(4, 2, 7, 2)
    b = A(2, -30, -2, 2)
    c = A(4, 7, 2, 2)
    d = A(2, 1, 12, -2)

    print(a == b)
    print(c == d)

main()
```

A.

```
True
False
```

B.

```
False
True
```

C.

```
True
True
```

D.

```
False
False
```

8. Which of the following choice is possible to be included in the answer of the program if the following program is executed by an interactive Python 3 interpreter?

```
def fun(n):
    if (n==1):
        return 1
    elif (n==2):
        return 2
    else:
        return fun(n-1)+fun(n-2)
print(fun(5))
```

A. 6

B. 8

C. 10

D. 12

9. Which of the following statement is wrong?

A. `NumPy` is the fundamental package for scientific computing with Python which contains writing data to files and vectorized computation.

B. If we want to create a null vector of size 10 but the fifth value which is 1,we can use the following code:

```python
import numpy as np
b = np.zeros(9) # init
b[4] = 1
```

C. If we want to reverse a vector (first element becomes last), we can use the following code:

```python
import numpy as np
c = np.zeros(227) # init
'''
Some operations here.
'''
c = c[::-1]
```

D. If we want to transform m which is a matrix into a 3×3 matrix with value from 0 to 8:

```python
import numpy as np
d = np.arange(9).reshape(3, 3)
```

10. Which of the following statement is wrong?

A. `Pandas` is an open source library providing high-performance, easy-to-use data structures and data analysis tools for Python.

B. The Pandas Series data structure is a one-dimensional, heterogeneous array with labels.

C. `set_index()` is a method to set a `list`, sequence or `dataframe` to the `dataframe` index in `Pandas`.

D. `Pandas` supports four types of multi-axis indexes, they are:

- `Dataframe.[]` is called the index operator;
- `Dataframe.loc[]` is used for labels;
- `Dataframe.iloc[]` is used for position or float based;
- `Dataframe.ix[]` is used for labels and integer-based.

11. Which of the following statement is correct?

A. We can ensure that a file object is properly closed after usage simply through using `with` statement.

B. Subclass can not use any method to access private attributes of the base class.

C. There can be no more than 1 `except` clauses in a `try` statement.

D. It is recommended to call class methods in this way: `object.f(p1,p2,...,pn)`.

12. Which of the statements is correct about the following program?

```python
class A:
    def __init__(self):
        self.calcI(30)
    def calcI(self, i):
        self.i = 2 * i

class B(A):
    def __init__(self):
        super().__init__()
        print("i from B is", self.i)
    def calcI(self, i):
        self.i = 3 * i

b = B()
```

A. The `__init__` method of only class B gets invoked.

B. The `__init__` method of class A gets invoked and it displays `i from B is 0`.

C. The `__init__` method of class A gets invoked and it displays `i from B is 60`.

D. The `__init__` method of class A gets invoked and it displays `i from B is 90`.

13. What is possible to be the output of the program below if executed by an interactive Python 3 interpreter?

```
def foo(str):
    s = ''
    try:
        x = int(str,2)
        s += "Succeed"
    except ValueError:
        s += "ValueError"
    else:
        s += "Else"
    finally:
        s += "Finally"
    return s
print(foo('110')+foo('120'))
```

A. `110+120`

B. `230`

C. `SucceedElseFinallyValueErrorFinally`

D. `SucceedFinallyValueErrorElseFinally`

14. What is possible to be the output of the program below if executed by an interactive Python 3 interpreter?

```
graph = {
        'a' : ['b', 'c'],
        'b' : ['a', 'c', 'd'],
        'c' : ['a','b', 'd','e'],
        'd' : ['b' , 'c', 'e', 'f'],
        'e' : ['c', 'd'],
        'f' : ['d']
        }
result = ''
def foo(graph, s):
    global result
    stack = []
    stack.append(s)
    seen = set()
```

```python
    seen.add(s)
    while len(stack) > 0:
        vertex = stack.pop()
        nodes  = graph[vertex]
        for node in nodes:
            if node not in seen:
                stack.append(node)
                seen.add(node)
        result += vertex
foo(graph, 'a')
print(result)
```

A. `abcdef`

B. `acedfb`

C. `bfdeca`

D. `fedcba`

15. What gets printed if the following program is executed by an interactive Python 3 interpreter?

```python
class BaseClass:
    def __init__(self):
        self.x = "BaseClass X"
        self.__x = "Private X"
    def __private(self):
        print("Private instance method in BaseClass")
    def public(self):
        print("Public instance method in BaseClass")
        self.__private()
        print(self.x)
class SubClass(BaseClass):
    def __init__(self):
        BaseClass.__init__(self)
        self.x = "SubClass X"
    def __private(self):
        print("Private instance method in SubClass")
    def public(self):
        BaseClass.public(self)
        self.__private()
        print(self.x)
s = SubClass()
s.public()
```

A.

```
Public instance method in BaseClass
Private instance method in BaseClass
SubClass X
Private instance method in SubClass
SubClass X
```

B.

```
Public instance method in BaseClass
Private instance method in BaseClass
BaseClass X
```

C.

```
Public instance method in BaseClass
Private instance method in BaseClass
BaseClass X
Private instance method in SubClass
SubClass X
```

D.

```
Public instance method in BaseClass
Private instance method in SubClass
SubClass X
Private instance method in SubClass
SubClass X
```

16. How many function calls (from Line 4 to 7) in the program below are valid (i.e. executed without exceptions)?

```
1 def foo(club,*member,**age):
2     pass
3 
4 foo('music','Alice','Bob')
5 foo('basketball','Alice','Bob','Alice'=12,'Bob'=18)
6 foo('dance',Alice=15,Bob=21)
7 foo('swim',Alice=9,Bob=15,'Alice','Bob')
```

A. 1

B. 2

C. 3

D. 4

17. What is the last line of the output of the following program?

```
a = {'a': 1 , 'b': 2 , 'c': 3, 'd': 4, 'e': 5, 'f': 6, 'g': 7}
for i in a.values():
    print(i)
    if i > 5:
        print('Exceed {}'.format(i))
        break
else:
    print('Loop Completely Finished.')
```

A. `5`

B. `6`

C. `Exceed 5`

D. `Exceed 6`

E. `Loop Completely Finished`

F. `4`

Section 2: Multiple Choices with No Less Than One Correct Answer

(3.5 pts * 10 = 35 pts) *Each question has* ***no less than one correct answer****. All the following programs are executed in an interactive* `Python3` *interpreter.*

Partial Scoring Notation:

- Partial Scoring will be **disabled** in Problem Set 17-21
- Partial Scoring will be **enabled** in Problem Set 22-26

18. Please select all the lines of codes which will cause errors in the following code segment.

```
1  x = 1
2
3  if x = 2:
4      a -= 2
5  else:
6      b = 9
7
8  if x == 1
9      return
```

A. Line 3

B. Line 4

C. Line 5

D. Line 6

E. Line 7

F. Line 8

19. Which of the following statements about file IO is/are true?

A. You can open files with Python function `open()` without `import`

B. If you don't provide the filename for `open()` function, it will automatically create a temp file under current directory

C. The default mode for `open()` function is `a`, since people usually append new contents to existing files rather than rewrite them

D. The file cursor moves while reading and you can manually set it to the location that you are interested in

20. Given the following function:

```python
def foo(x, y = 1, *args, **kwargs):
    pass
```

Which of the following function call(s) could be properly executed by an interactive Python 3 interpreter?

A. `foo(2)`

B. `foo(x=2, 2)`

C. `foo(2, y=2)`

D. `foo(2, 2, 2)`

E. `foo(2, 2, x=3, y=3)`

F. `foo(2, s=3, t=3)`

21. Which of the following statement(s) is/are true?

A. the `dtype` of `array([1, 2.0, 0j])` is either `complex64` or `complex128`.

B. the `ndarray` can't store multiple type data, while list can.

C. Assume an `ndarray` called `arr`, if `arr.base == None`, then `arr` don't have its own memory.

D. We can learn utilities (e.g. input arguments, output arguments, etc.) of `NumPy` functions by `help()`.

22. Which of the following statements about iterators and generators is/are correct?

A. All generator objects are iterators.

B. All iterables are iterators.

C. Generator objects cannot be created without using `yield` statements.

D. Local variables and the execution start is automatically saved between calls using generators.

E. An iterator must support a method called `__next__()`.

23. Which of the following expression(s) might be able to create a transpose of the given matrix?

```python
matrix = [
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12]
]
```

(Hint: The result should be `[ [ 1, 5, 9 ], [2, 6, 10 ], [3, 7, 11], [4, 8, 12] ]`.)

A.

```python
[ row[ col ] for row in matrix for col in range(4) ]
```

B.

```
[ [ row[ col ] for row in matrix ] for col in range(4) ]
```

C.

```
[ row[ col ] for col in range(4) for row in matrix ]
```

D.

```
[ [ row[ col ] for col in range(4) ] for row in matrix ]
```

E.

```
[ [matrix[i][j] for i in range(3)] for j in range(4) ]
```

F.

```
[ [matrix[i][j] for j in range(4)] for i in range(3) ]
```

24. Which of the expression(s) is/are illegal in Python 3?

A.

```
a,b = 1,2
b == not a
```

B.

```
None == False | 1.0 is not 1
```

C.

```
"python" + 3 + "ic"
```

D.

```
"python" * 3 + "ic"
```

E.

```
"python3"[6] = 2
```

25. Consider the following class that implements some basic vector operations, then choose the *false* statement(s)

```
class Vector:
    def __init__(self, l):
        self.li = l
        self.length = len(l)

    def __add__(self, other):
```

```python
        if self.length != other.length:
            return
        else:
            tmp = [0] * self.length
            for i in range(self.length):
                tmp[i] = self.li[i] + other.li[i]
            return tmp

    def __mul__(self, other):
        if self.length != other.length:
            return
        else:
            tmp = 0
            for i in range(self.length):
                tmp += (self.li[i] * other.li[i])
            return tmp
```

A. The output for `print(Vector([1, 2]) + Vector([3, 4]))` is `[4, 6]`.

B. The output for `print(Vector([1, 2, 3]) * Vector([3, 4]))` is `NULL`.

C. `Vector` objects can be initialized with lists, tuples and sets without incurring exceptions in further operations, but not dictionaries.

D. Both `__add__()` and `__mul__()` functions are private methods and both of them could not be called by other objects.

26. Suppose that we want to filter a list `a = [1,2,2,3,4,4,5]` so that it only contains odd elements, i.e., `a = [1,3,5]` after doing operations on it. Which of the following codes can *not* satisfy our need?

A.

```python
a = [1,2,2,3,4,4,5]
for i in a:
    if i % 2 == 0:
        del i
```

B.

```python
a = [1,2,2,3,4,4,5]
for i in a:
    if i % 2 == 0:
        a.remove(i)
```

C.

```python
a = [1,2,2,3,4,4,5]
b = a
for i in b:
    if i % 2 == 0:
        a.remove(i)
```

D.

```python
a = [1,2,2,3,4,4,5]
b = []
for i in a:
    if i % 2:
        b.append(i)
a = b
```

27. Consider the following class that mimics forming a triangle in the real word, then choose the *false* statement(s)

```python
class Triangle:
    def __init__(self, a, b, c):
        try:
            if(a <= 0 or b <= 0 or c <= 0):
                raise ValueError("The lengths of edges must be positive")
            else:
                try:
                    if((a+b)<=c or (a+c)<=b or (b+c)<=a):
                        raise ValueError("Cannot form a triangle")
                    else:
                        self.a, self.b, self.c = a, b, c
                except BaseException:
                    raise Exception("Something caused forming exception")
        except ValueError as e:
            print(e, end='. ')
        else:
            print("Triangle formed", end='. ')
        finally:
            print("Exiting constructor.")
```

A. The output for `Triangle(1, 1, 1)` is `Triangle formed. Exiting constructor.`

B. The output for `Triangle(0, 1, 1)` is `Cannot form a triangle. Exiting constructor.`

C. The output for `Triangle(1, 1, 2)` is `Cannot form a triangle. Exiting constructor.`

D. The output for `Triangle()` is `Something caused forming exception. Exiting constructor.`

Section 3: Blank Filling

(5 pts + 12 pts + 12 pts = 29 pts)

28. Grader

(5 pts)

Mr. Suica wants to give a quiz to his students. Just like our quizzes and this exam, there are blank filling questions which require students to fill in their code. However, there can be multiple answers for a blank (for example, whitespaces at minor places can be added or ignored, `a = 1` is equivalent to `a=1`). To deal with this, Mr. Suica wants to use a regular expression to match as many possible correct answers as possible.

For example, a question requires the students to write a Python expression to check if the given `distance` is more than 100 kilometers. The number `100` can be included or excluded. In addition, it can be `int` or `float`. If it is a `float`, there must be a decimal dot, and there can be zero or more `0`s after the dot.

The answers like the following expressions are allowed:

```
distance > 100
distance > 100.
distance>=  100
distance  >=100.0000
```

Write a regular expression such that it could match all the above and all the other acceptable answers.

Note:

- Your answer should be only **one** line.
- There are several kinds of white spaces that Python support. Use `\s` here to refer to one whitespace.

29. Zip

(12 pts)

In the previous discussion class, we introduced the built-in class `zip`. It accepts multiple iterables, and makes an iterator that aggregates (聚集) elements from each of the iterables. For example, `zip([1, 2, 3], [4, 5, 6])` is an iterator which produces the sequence `(1, 4), (2, 5), (3, 6)`.

However, `zip()` will stop as long as the shortest iterable ends. For example, `zip([1, 2], [4, 5, 6])` produces the sequence `(1, 4), (2, 5)`. It ignores the trailing `6` of the second iterable, because the first iterable has ended.

In order to save those trailing values, **we implement a class `LongZip` which stops only when all the iterables end**. For the iterables which end earlier, we fill in **dummy values** to accompany (伴随) trailing values. For example, if the dummy value is `0`, `LongZip([1, 2], [4, 5, 6], fill=0)` should produce the sequence `(1, 4), (2, 5), (0, 6)`.

Complete the following program. The test case at the end of the program must pass.

```
class LongZip:
    def __init__(self, *iters, fill=None):
        self.iters = [_______(A)_______ for it in iters]
        self.remain = len(self.iters)
        self.fill = fill

    def __iter__(self):
        return self

    def __next__(self):
        if self.remain == 0:
            raise StopIteration
        values = []
        for i in range(len(self.iters)):
```

```python
            if self.iters[i] is None:
                values.append(self.fill)
            else:
                try:
                    value = _______(B)_______
                except StopIteration:
                    _______(C)_______
                    if self.remain == 0:
                        raise StopIteration
                    values.append(self.fill)
                    self.iters[i] _______(D)_______
                else:
                    values.append(value)
        return _______(E)_______

assert list(LongZip([1, 2], [4, 5, 6], fill=0)) == [(1, 4), (2, 5), (0, 6)]
```

Note:

- Please fill in your answers in the answer table in the Answer Book.

30. Number System Conversion

(12 pts)

The following function is doing number system conversion, the maximum base the function support is 36. Number system conversion is the process to convert numbers between different base. In this case, we first convert the number to decimal (十进制的) representation and then convert it to the target base. For example, converting octal (八进制的) number `1234` to its hexadecimal (十六进制的) representation `29C` is done by following:

1. $1234_{oct} = 1 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 668$
2. $668 = 2 \times 16^2 + 9 \times 16^1 + 12 \times 16^0 = 29C_{hex}$ The coefficient `2,9,12` could be obtained by following:

$668 \div 16 = 41 \cdots\cdots 12$

$41 \div 16 = 2 \cdots\cdots 9$

$2 \div 16 = 0 \cdots\cdots 2$

Input parameters of function `value2digit`: `value` is an integer representing the true value of a digit.

Return value of function `value2digit`: A character representing the digit under certain base.

Input parameters of function `digit2value`: `digit` is a character representing a digit.

Return value of function `digit2value`: An integer representing the true value of the digit under certain base.

Input parameters of function `Convert`:

- `input_number` is a string object formed by numbers (`0-9`) and uppercase letters (`A-Z`) like `"123F"` and `"SI100"`. Assume it is not empty.
- `input_base` is a number denoting the base of the input. It should have `2 <= input_base <= 36`

- `output_base` is a number denoting the base of the output. It should have `2 <= output_base <= 36`

Return value of function `Convert`: A string of the number after conversion. It should only contain numbers (`0-9`) and uppercase letters (`A-Z`).

Part of the code is already given. You need to fill in the blanks such that the program could work properly.

```python
def value2digit(value: int) -> str:
    if value < 10:
        _____(A)_____
    return chr(value-10+ord("A"))


def digit2value(digit: str) -> int:
    if _____(B)_____:
        return ord(digit)-ord("A")+10
    return int(digit)


def Convert(input_number: str, input_base: int, output_base: int) -> str:
    decimal_value = 0
    input_base_shift = _____(C)_____
    for digit in input_number[::-1]:
        decimal_value += digit2value(digit)*input_base_shift
        _____(D)_____

    output = ""
    while decimal_value > 0:
        output = _____(E)_____
        decimal_value //= output_base

    return output
```

Note:

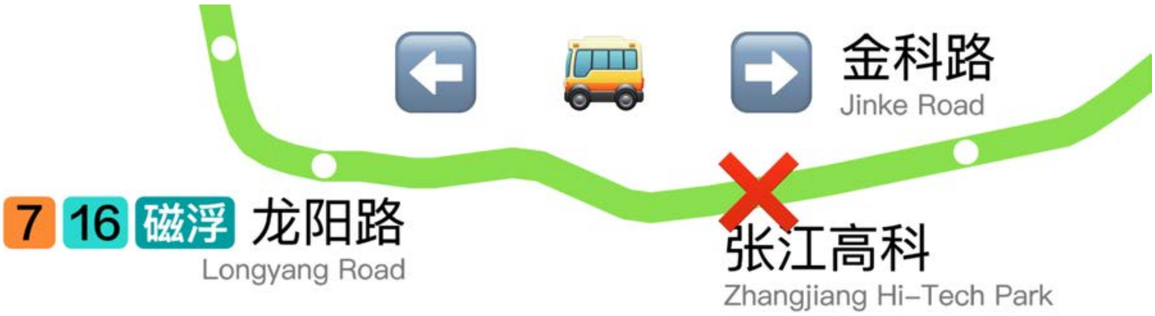
- Please fill in your answers in the answer table in the Answer Book.

Section 4: Integrated Problems

(12 pts + 5 pts + 8 pts + 14 pts + 5 pts = 44 pts)

31. Simple Shuttle Bus

(12 pts)



In 2010, Zhangjiang High-Tech Park Station (张江高科站) of Metro Line 2 (轨交 2 号线) was shut down for two weeks. To keep the traffic flow, the metro company decided to arrange shuttle buses to transport passengers between Longyang Road Station (龙阳路站) and Jinke Road Station (金科路站), which are both adjcent on Line 2 to Zhangjiang High-Tech Park Station.

In this task, you are going to implement a simplified program to simulate this scene which is similar to HW1 task1. Assume we only have one shuttle bus to transport passengers between Longyang Road Station and Jinke Road Station. The bus's **capacity is 3** and it takes **1200 seconds** to go from one station to another (a.k.a. **round trip time**). Bus only leaves when it's **full**. When the bus leaves a station, we need to record its departure time and passengers' name list.

The input of the program includes three lists: The list called `boarding_info` records which station the bus is at and the time it arrives the station. For example, `boarding_info = ["longyang_road", 0]` indicates the bus is now at Longyang Road station and it arrives there at time 0.

Besides, `longyang_road_passenger_list` is the passenger information in Longyang Road Subway Station and `jinke_road_passenger_list` is the passenger information in Jinke Road Station. The list above consists of tuples, which has only two element each. For each tuple, the first element is passenger name in string and second one is passenger's arrival time in integer. Example of input format:

```
jongyang_road_passenger_list = [("WangDaChui", 0), ("GeZiWang", 400), ("WangXiaoMing", 600)]
jinke_road_passenger_list = [("ShaoKaoDaWang", 0), ("XiYangYang", 900), ("ManYangYang", 2000)]
```

The output of the program is a dictionary and its keys are the departure time in integer and values are the corresponding list of passengers' name. Example of output format (since it's dictionary, the order of the **key** doesn't matter):

```
{600: ['WangDaChui', 'GeZiWang', 'WangXiaoMing'], 2000: ['ShaoKaoDaWang', 'XiYangYang', 'ManYangYang']}
```

Tips: The passenger's number at each station is always a multiple of bus's capacity. So, there is no need to consider the situation when only one or two passengers board the bus.

(4 pts) Find errors in your program

Following is the given information:

```python
boarding_info = ["Longyang Road", 0]
longyang_road_passenger_list = [("WangDaChui", 0), ("GeZiWang", 400), ("WangXiaoMing", 600), ("XiaoKeAi", 610), ("JiLingGui", 700), ("MeiTouNao", 800)]
jinke_road_passenger_list = [("ShaoKaoDaWang", 0), ("XiYangYang", 900), ("ManYangYang", 2000)]
```

Below is the code to implement the scheduling algorithm. The output information is stored in the variable called **result**. However, there are **3** errors in the code. For each error, indicate the line number, briefly explain what the error is, and correct the statements.

```python
 1  bus_capacity = 3
 2  round_trip_time = 1200
 3  result = {}
 4  while len(longyang_road_passenger_list) != 0 or len(jinke_road_passenger_list) != 0:
 5  depature_time = 0
 6      boarded_passengers = None
 7      if boarding_info[0] == "Longyang Road":
 8          for curr_passenger in range(bus_capacity):
 9              passenger_tuple = longyang_road_passenger_list.pop(0)
10              boarded_passengers.append(passenger_tuple[0])
11              depature_time = passenger_tuple[1]
12          if len(result) != 0 and boarding_info[1] > depature_time:
13              depature_time = boarding_info[1]
14          result[depature_time] = boarded_passengers
15          boarding_info = ["Jinke Road", depature_time+round_trip_time]
16      else:
17          for curr_passenger in range(bus_capacity):
18              passenger_tuple = jinke_road_passenger_list.pop(0)
19              boarded_passengers.append(passenger_tuple[0])
20              depature_time = passenger_tuple[1]
21          if len(result) != 0:
22              depature_time = boarding_info[1]
23          result[depature_time] = boarded_passengers
24          boarding_info = ["Longyang Road", depature_time+round_trip_time]
```

(4 pts) Fill in the blanks

The program is then polished by using classes and functions. Similarly, the output information is stored in the variable called **result**. You are required to fill in the blank lines so that the program will output correct result.

```python
 1  class ShuttleBus:
 2      def __init__(self, longyang_road, jinke_road):
 3          self.passengers_at_longyang = longyang_road
 4          self.boarding_info = ['Longyang Road Station', 0]
 5          self.bus_capacity = 3
 6          self.passengers_at_jinke = jinke_road
 7          self.result = {}
 8          self.round_trip_time = 1200
 9          while len(self.passengers_at_longyang) != 0 or len(self.passengers_at_jinke) != 0:
10              self.process()
11          print(self.result)
```

```python

    def process(self):
        departure_time = 0
        boarded_passengers = []
        if self.boarding_info[0] == 'Longyang Road Station':
            data_source = self.passengers_at_longyang
            self.boarding_info[0] = 'Jinke Road Station'
        else:
            data_source = self.passengers_at_jinke
            self.boarding_info[0] = 'Longyang Road Station'

        for i in range(self.bus_capacity):
            passenger_tuple = data_source.pop(0)
            boarded_passengers.append(______(A)_______)
            departure_time = passenger_tuple[1]

        # check if bus is on the way
        bus_arrive_time = ______(B)_______
        if len(self.result) != 0 and bus_arrive_time > departure_time:
            departure_time = bus_arrive_time
            self.boarding_info[1] += self.round_trip_time
        else:
            self.boarding_info[1] = departure_time

        # load result
        _______(C)_______


if __name__ == "__main__":
    longyang_road_passenger_list = [("WangDaChui", 0), ("GeZiWang", 400), ("WangXiaoMing", 600), ("XiaoKeAi", 610),("JiLingGui", 700), ("MeiTouNao", 800)]
    jinke_road_passenger_list = [("ShaoKaoDaWang", 0), ("XiYangYang", 900), ("ManYangYang", 2000)]
    bus_record = ShuttleBus(longyang_road_passenger_list, jinke_road_passenger_list)
```

The program should output:

```
{600: ['WangDaChui', 'GeZiWang', 'WangXiaoMing'], 2000: ['ShaoKaoDaWang', 'XiYangYang', 'ManYangYang'], 3200: ['XiaoKeAi', 'JiLingGui', 'MeiTouNao']}
```

(4 pts) Debug with `stdin` and `stdout`

You are debugging the implementation in the last section. Suppose the input and output handling part of this task has been finished by the TAs. The input and output part are similar to the one your encountered in HW1. Recall what you have implemented in HW1, test cases are stored in a file and you are redirecting the content in the file to the standard input of your program.

Since the commands for those operations are platform-specific, please first write down the operation system (and optionally the shell) you are using: `__________` (0 pt) (possible answer includes `Linux`, `macOS`, `Windows with powershell`, `Windows with cmd.exe` and so on). We will grade your answer to the following questions based on your answer to this blank.

Note: you are required to use Python 3 in all your homework assignments and your Python interpreter is accessible with the name `python3`.

Now, you are reading in the test case from the file `testcase.in` in your current working directory. You are not redirecting the output of your program in this question. Suppose your program is called `shuttle_bus.py`, write your full command for redirecting the file content of `testcase.in` to `stdin` of your program below next to the command prompt (命令提示符) `#`

First command:

```
# ________________________________________________________________________________
```

Now you are saving the output (in `stdout`) of your `shuttle_bus.py` to a file called `testcase.out` while read in the test case from `testcase.out`. Write your full command for finishing this task next to the command prompt `#`

Second command:

```
# ________________________________________________________________________________
```

Now you are comparing the content of `testcase.out` to the reference output in `testcase.ref`. Since the output file is very large and it is almost impossible to compare them with your eyes, you are using some tools provided by your shell to accomplish this task. Write your full command for comparing `testcase.out` with `testcase.ref` below next to the command prompt `#`

Third command:

```
# ________________________________________________________________________________
```

32. Left Outer Join

(5 pts)

You should be familiar with the `join` operation in database since you have written a simple database in Homework 4. What the `join` operation do is basically combining two (or more) tables together. In fact, there are many types of join operations. The one you encountered in Homework 4 is the simplest one called inner join. You are asked to implement another kind of join operation in this question called **left outer join**.

The **left outer join** is very similar to the inner join in HW4. The only difference is that, say we have two tables A and B, **left outer join** will preserve all the rows in A (the table appears on the left) even if no matching row (i.e. row with the same primary key) could be found in B.

The following is an example. In `student.csv` we have several students with their student ID as the primary key. `gpa.csv` has the GPA of some students in `student.csv`.

`student.csv`

id	name	school	major
23456123	Li Xiaoming	SIST	CS
45280742	Ge Ziwang	SPST	PHY
12567923	Wang Dachui	SLST	BIO

`gpa.csv`

id	gpa
23456123	3.98
45280742	3.76

After **left outer join** on `student.csv` and `gpa.csv`, the result table will still have 3 rows. But the GPA of Wang Dachui is missing because we could not find it in `gpa.csv`. (Notice that in inner join, you are expected to produce only 2 rows in the result table)

student.id	student.name	student.school	student.major	gpa.gpa
23456123	Li Xiaoming	SIST	CS	3.98
45280742	Ge Ziwang	SPST	PHY	3.76
12567923	Wang Dachui	SLST	BIO	`None`

Fill in the blanks [A], [B] and [C] in the code snippet below so that the function `get_left_outer_join()` could produce the **left outer join** of table `left` and `right` correctly (you are not asked to do filtering and any other things as required in HW4) and return the result as a list of rows. `left` and `right` are both of `Table`-type as in HW4. You are going to placehold the fields with no value with `None` in your result.

```python
from yoursql import Table, Row
from itertools import chain
from typing import List

def get_left_outer_join(left: Table, right: Table) -> List[Row]:
    result = []

    lname = left.get_table_name()[:-4]
    rname = right.get_table_name()[:-4]
    primary_key = left.get_primary_key()

    for lrow in left:
        matched = False
        for rrow in right:
            if lrow[lrow.get_primary_key()] == rrow[rrow.get_primary_key]:
                this_keys = chain(['.'.join([lname, key]) for key in lrow.keys()],
                                  ['.'.join([rname, key]) for key in lrow.keys()])
                this_data = chain([lrow[key] for key in lrow.keys()],
                                  [lrow[key] for key in lrow.keys()])
                result.append(
                    Row(this_keys, this_data, primary_key='.'.join([lname,
primary_key])))
                ______(A)_______
                break
        if not matched:
            this_keys = _______(B)_______
            this_data = _______(C)_______
            result.append(Row(this_keys, this_data,
                              primary_key='.'.join([lname, primary_key])))

    return result

if __name__ == '__main__':
    students = Table('student.csv')
    gpa = Table('gpa.csv')
    print(get_left_outer_join(students, gpa))
```

33. Bad Table

(8 pts)

Wang Dachui implemented his `Table()` object in HW4. He is struggling with this implementation because he could not pass some test cases in the Auto-grader. Find the problem in his code that may cause his code to crash or get unexpected results for some test cases. Suppose other parts of Wang Dachui's code are correct.

There are in total 5 mistakes in the following program. One of the mistakes has been pointed as an example. Please answer the question in the format of:

Line No.	Your Explanation	Your Correction
6	redundant white space before the statement	remove a white space before the statement of Line 6

and fill in your answers in the given table in the Answer Book

```python
1  class Table():
2  
3      def __init__(self, filename, rows=None, keys=None, primary_key=None):
4          self.__table_name = filename
5          if isinstance(rows, list):
6               self.__rows = list(rows)
7              self.__keys = rows[0].keys()
8              self.__pk = rows[0].get_primary_key()
9          else:
10             with open(filename) as f:
11                 self.__keys = [key.strip()
12                                for key in f.readline().strip().split(',')]
13                 self.__pk = primary_key if primary_key else (
14                     self.__keys[0] if len(self.__keys) else None)
15                 if not self.__pk in self.__keys:
16                     raise KeyError
```

```
                rows_text = f.readlines()
                self.__rows = []
                for row_text in rows_text:
                    row = [field.strip()
                           for field in row_text.strip().split(',')]
                    if len(row) != 0:
                        self.__rows.append(Row(self.__keys, row, self.__pk))
        self.__keys = sorted(self.__keys)

    def __iter__(self):
        return self

    def __next__(self):
        self.__iter_idx += 1
        if self.__iter_idx >= len(self.__rows):
            raise StopIteration
        return self.__rows[self.__iter_idx]

    def __getitem__(self, key):
        for row in self.__rows:
            if row[self.__pk] == key:
                return row
        raise ValueError

    def __len__(self):
        return len(self.__rows)

    def keys(self):
        return self.__keys.copy()

    def get_primary_key(self):
        return self.__pk

    def get_table_name(self):
        return self.__table_name

    def export(self, filename=None):
        fn = filename if filename else self.__table_name
        with open(fn) as f:
            f.write('{}\n'.format(','.join(self.__keys)))
            for r in self.__rows:
                f.write('{}\n'.format(','.join([str(r[k]) for k in r])))
```

34. Determinant and Inverse

(14 pts)

Determinant and inverse are 2 fundamental operations in Linear Algebra for matrices. Both of the two operations have implemented by you in Homework 5. This problem set will give you a brief view of them.

Note:

- In this problem set, `NumPy` is imported via:

```python
import numpy as np
```

- Remember to write your answers down in the required section of the Answer Sheet. Writing down in a wrong section will be graded as 0 point for related problem sets.

Determinant (行列式)

1. (2 pts) Is recursion used in determinant calculation? If so, how is recursion used during determinant calculation?

   Note: Only answer `Yes` or `No` or failing to explaining where recursion is used will get 0 point.

2. (4 pts) The following function `determinant()` is a function to calculate determinant implemented by Mr. Sailboat. Please fill in the 3 blanks in order to make it work properly.

   Clarification on Input and Output:

   - **Input**:
     - `mat_list`: `list`, size: $n \times n$. Each element in `mat_list` represents each row of the matrix
   - **Output**:
     - `res`: `float`. Determinant of `mat_list`

   Note:

   - Assume all the inputs are valid. In the other word, no exception will be raised if the function is properly implemented

```python
def determinant(mat_list):
    if len(mat_list) == _____(A)_____:
        _______(B)_______
    else:
        res = 0.0
        for numi in range(len(mat_list)):
            Matx = []
            for numj in range(len(mat_list)-1):
                Matx.append([])
                for numk in range(len(mat_list)):
                    if numi != numk:
                        Matx[numj].append(mat_list[numj+1][numk])
            ___________(C)___________
        return res
```

3. (2 pts) Assume all 3 blanks of `determinant()` are filled in correctly, what gets printed if the following program is executed?

```python
mat_list = [[2, 0, 2], [2, 3, -1], [0, 1, 0]]
res = '%.4f' % determinant(mat_list)
print(res)
```

Inverse (逆矩阵)

4. (1 pt) Which of the following `NumPy` method/function will be used to calculate inverse matrix?

   Hint: Recall Task 1.2 of Homework 5

   A. `np.inv()`

B. `np.linalg.inv()`

C. `np.inverse()`

D. `np.linalg.inverse()`

5. (1 pt) In a Kalman Filter, which part will involve inverse matrix calculation?

   Hint: Recall Task 1.2 and 6 of Homework 5

   A. Reset

   B. Prediction

   C. Calibration

   D. Update

6. (4 pts) The following program is implemented to calculate inverse matrix. There are **3** mistakes in the following program. You need to first pointing out where mistakes are by pointing out line numbers. Then you need to explain why the line is wrong and how to fix those errors by rewriting those lines of code. Notice that you could only rewrite **one line** for each mistake.

   One mistake has been given as an example.

   Note:

   - Assume all the inputs are valid. In the other word, no exception will be raised if the function is properly implemented
     - Assume all the input matrices are invertible
   - Function `detminant()` is the same function in Problem Set 3. Assuming `detminant()` has been properly implemented

```
 1  def MatTrans(mat_list):
 2      return [[mat_list[a][b] for b in range(len(mat_list))] for a in range(len(mat_list[0]))]
 3
 4  def MatDiv(mat_list, divNum):
 5      return [[mat_list[rowNum][colNum]/divNum for colNum in range(len(mat_list[0]))] for rowNum in range(len(mat_list))]
 6
 7  def inv(mat_list):
 8
 9      n = len(mat_list)
10
11      detx = []
12      if n == 2:
13          detx.append([])
14          detx.append([])
15          detx[0] = [mat_list[1][1], -mat_list[1][0]]
16          detx[1] = [-mat_list[0][1], mat_list[0][0]]
17      else:
18          for numi in range(n):
19              detx.append([])
20              for numj in range(n):
21                  detxx = []
22                  x = -1
23                  for numk in range(n):
24                      if numk != numi:
25                          x += 1
26
```

```
27                          detxx.append([])
28                          for nump in range(n):
29                              if numj != nump:
30                                  detxx[x].append(mat_list[numk][nump])
31
32                  detx[numi].append(determinant(detxx) * (-1) **
    (numi+numj+1))
33
34      detx_trans = MatTrans(detx)
35      detx_det = determinant(mat_list)
36      res = MatDiv(detx_trans, detx_det)
37
38      return res
```

Answer Example:

Line Number	Description of the Mistake	Your Correction for the Mistake
15	Inverse calculation for 2×2 matrices incorrect	`detx[0] = [mat_list[0][1], -mat_list[1][0]]`

35. Tic-Tac-Toe

(5 pts)

Tic-tac-toe is a paper-and-pencil game for two players, `x` and `o`, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

There are **4** mistakes in the code (**including** the given example). For each mistake, indicate the line number, briefly explain what the mistake is, and correct the statements.

Line No.	Your Explanation	Your Correction
18	Syntax Error! No `else if` in Python!	`elif player == 'x':`

Hints:

- `&`, `|`, `<<` are bitwise operators.
- For the first element `7` in `winCondition`, we can write it in binary as `0b000000111` which means that there is 3 marks in the first horizontal row.
- Another example, if a chess board is:

```
 | |
-+-+-
 | |
-+-+-
 | |o
```

It will be expressed in binary as `0b100000000`.

```
boardO, boardX = 0, 0


def printBoard(boardO, boardX):
    board = [[' '] * 3] * 3
    for i in range(3):
        for j in range(3):
            if boardO & (1 << (i*3+j)):
                board[i][j] = 'o'
            elif boardX & (1 << (i*3+j)):
                board[i][j] = 'x'
    print('\n-+-+-\n'.join(['|'.join(line) for line in board]))


def updateBoard(i, j, player):
    if player == 'o':
        boardO |= (1 << (i*3+j))
    else if player == 'x':
        boardX |= (1 << (i*3+j))


def getResult(boardO, boardX):
    winCondition = [7, 7 << 3, 7 << 6, 73, 73 << 1, 73 << 2, 233, 84]
    for condition in winCondition:
        if boardO & condition == condition:
            print('o wins!')
            return
        elif boardX & condition == condition:
            print('x wins!')
            return
    print('Draw!')


updateBoard(1, 1, 'o')
updateBoard(1, 2, 'x')
printBoard(boardO, boardX)
print('='*20)
updateBoard(0, 1, 'o')
updateBoard(2, 1, 'x')
updateBoard(0, 0, 'o')
updateBoard(0, 2, 'x')
updateBoard(2, 2, 'o')
printBoard(boardO, boardX)
getResult(boardO, boardX)
'''
Output:
 | |
-+-+-
 |o|x
-+-+-
 | |
====================
o|o|x
-+-+-
 |o|x
-+-+-
 |x|o
o wins!
```

```
'''
```

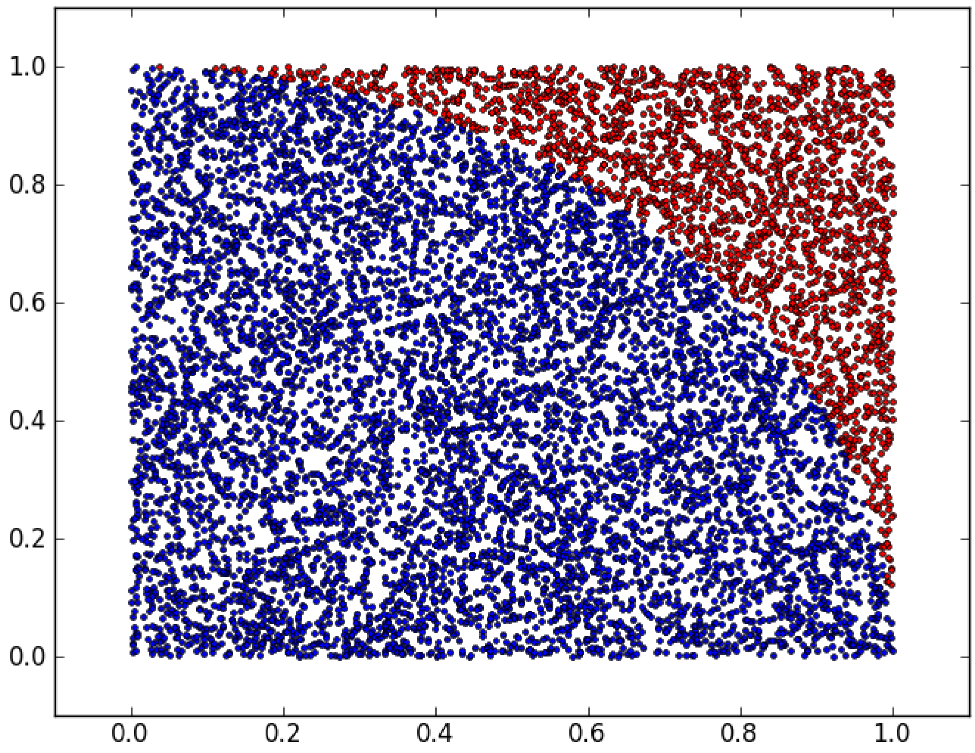
Section 5: Bonus

(4 pts)

36. Monte Carlo Estimation

(2 pts * 2 = 4 pts)

You are estimating the value of the math constant π using the Monte Carlo method. Monte Carlo method requires you to cast n points onto a square with side length (边长) 1 and one of whose vertex (顶点) being at $(0,0)$, and count the number (denote as m) of points whose distance with $(0,0)$ is less than 1 (i.e., in the quarter circle (四分之一圆)). The estimate value of π is then given by $\pi = \frac{4m}{n}$. Fill in the blank in the following code so that it could produce the correct estimation of π (this method is called something like 投针法估算π值 in your high school mathematic textbook).



```
from numpy import random, sqrt

# Ask the user how many points he/she want to cast?
n = int(input("Please tell me how many points you want to cast? "))

# generate 2 * n numbers that fall in [0, 1], each column represents a point.
pts = random.rand(n,2)

# Count the number of points fall in the circle.
pts_in = sqrt(pts[:,0]**2+pts[:,1]**2) __________(A)__________
m = __________(B)__________
```

```python

# calculate the value of pi.
pi = (4 * m) / n

print("The value of pi estimated by the Monte Carlo method using {} points is {}".format(n, pi))
```

END OF THE FINAL EXAM