Final Exam

2023.06.15 (Thursday) 10:10 – 12:00

1. **[Python Basics, Conditionals & Loops]** In each of the following questions, you are asked to show what will be printed out. If there is an error, please explain why it is an error. (14%)

(a)
```
s = 1.23
print(int(float(str((s))))+int(s))
```

(b)
```
_, __, ___ = 2, 3, 5
x, y, z = _*__, __*___, ___*_
print(x, y, z)
```

(c)
```
x, y = 5, 0
while x != y:
   x, y = x - 1, y + 2
print(x, y)
```

(d)
```
a, b = 72, 120
while b != 0
    t = b
    b = a % b
    a = t
print(a)
```

(e)
```
score = 20
while score > 1:
   score /= 2
print(score)
```

(f)
```
s = "yadyam evol i"
print(s.split()[0][::-1])
```

(g)
```
n = 8
a, b, i = 1, 1, 3
while i:
   a, b = b, a + b
   i = i - 1
print(a)
```

2. **[Lists]** In each of the following questions, you are asked to show what will be printed out. If there is an error, please explain why it is an error. (24%)

(a)
```
my_list = [1, 2, 3, 4, 5]
print(my_list[1::2])
```

(b)
```
my_list = [[1, 2], [3, [4, [5, 6]]]]
print(my_list[-1][-1][-1][0])
```

(c)

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
flat_list = []
for sublist in nested_list:
    for item in sublist:
        flat_list.append(item)
print(flat_list)
```

(d)

```
numbers = [-1, 5, -3, 9, 0, -11, 4]
positives, negatives = [], []
for num in numbers:
    if num > 0:
        positives.append(num)
    elif num < 0:
        negatives.append(num)
    else:
        break
print(positives, negatives)
```

(e)

```
values = [2, 0, 2, 3]
for v in values:
    v = 2 * v
print(v)
```

(f)

```
a = [1, 2, 3, 4, 5]
for i in range(len(a), -1, -1):
    print(a[i], end=" ")
```

(g)

```
days = [['Tue','May',9], ['Thu','May',11], ['Fri','Jun',30]]]
print(days[0][-1])
```

(h)

```
l = [1, 2, 3]
l.extend([4, 5, 6])
l.append([7, 8, 9])
print(l)
```

(i)

```
l = ['n', 'b', 'a']
for i in range(len(l)):
   for j in range(i+1, len(l)+1):
      print(sorted(l[i:j]))
```

(j)

```
nums = [1, 2, 3, 4, 5, 6, 7]
k = 4
n = len(nums)
k %= n
nums[:] = nums[n-k:] + nums[:n-k]
print(print(nums))
```

(k)

```
a, b = 15, 39
small = a if a > b else b
cf, pf = [], 1
while pf <= small:
   if a % pf == 0 or b % pf == 0:
      cf.append(pf)
   pf += 1
print(cf)
```

(l)

```
nums = [1, 3, 5, 7, 4, 6, 8, 10, 9, 11]
results = []
for i in range(len(nums)-1):
   if nums[i] % 2 == 0:
      continue
   if nums[i+1] % 2 != 0:
      results.append([nums[i], nums[i+1]])
print(results)
```

3. **[Dictionary & Tuple]** In each of the following questions, you are asked to show what will be printed out. If there is an error, please explain why it is an error. (18%)

(a)

```
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
for key in d:
    if d[key] > 2:
        for i in range(d[key]):
            print(key, end='')
print()
```

(b)

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
d = dict(zip(keys, values))
print(d)
```

(c)

```
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
for key in list(d.keys()):
    if d[key] % 2 == 0:
        del d[key]
print(d)
```

(d)

```
stu = [("Joy", "A"), ("Bob", "B"), ("Eva", "B"), ("Fox", "A")]
grades = {"A": [], "B": []}
for student in stu:
    grades[student[1]].append(student[0])
print(grades)
```

(e)
```
my_tuple = (1, 2, 3)
my_tuple[0] = 0
print(my_tuple)
```

(f)
```
dict_a = {"name": "Tom", "age": 30, "city": "LA"}
dict_b = {"name": "Joy", "age": 25, "city": "Paris", "job": "Art"}
dict_a.update(dict_b)
print(dict_a)
```

(g)
```
data = [("A","X"), ("A","Y"), ("B","X"), ("B","Y"), ("B","X")]
crosstab = {}
for item in data:
    row, col = item
    if row not in crosstab:
        crosstab[row] = {}
    if col not in crosstab[row]:
        crosstab[row][col] = 0
    crosstab[row][col] += 1
print(crosstab)
```

(h)
```
d = {"a": 1, "b": 2, "c": 3}
reversed_dict = []
for key, value in d.items():
    reversed_dict[value] = key
print(reversed_dict)
```

(i)
```
original_dict = {'a': 1, 'b': 2, 'c': 3}
new_dict = {}
for i, key in enumerate(original_dict.keys()):
    new_dict[key] = i
print(new_dict)
```

4. **[Files & Exception Handling]** In each of the following questions, you are asked to show what will be printed out. If there is an error, please explain why it is an error. (16%)

(a)
```
file_name = "test.txt"
f = open(file_name, "r")
t1 = f.readlines()
f.close()
print(t1)
```

go.txt contains three lines:
```
Merry Christmas
New Year
Winter Vacation
```

(b)
```
file_name = "go.txt"
f = open(file_name, "r")
t1 = f.readlines()
f.close()
print(t1)
```

(c)

```
file_name = "go.txt"
f = open(file_name, "r")
t1 = f.read()
print(t1)
f.seek(0)
t2 = f.readline()
f.close()
print(t2)
```

(d)

```
mylist = []
with open("go.txt", "r") as f:
    for line in f:
        mylist.append(line.strip())
print(mylist)
```

(e)

```
with open("test_file.txt", "w") as file:
    file.write("Hello, World!")
with open("test_file.txt", "r") as file:
    print(file.read())
```

(f)

```
x, y, z = 5, 2, 0
try:
    result1 = x / y
    result2 = x / w
    result3 = x / z
except ZeroDivisionError:
    print("Divided by zero")
except NameError:
    print("Variable is not defined")
except:
    print("Other errors")
```

(g)

```
x, y, z = 5, 2, 0
try:
    result1 = x / y
    result3 = x / z
    result2 = x / w
except NameError:
    print("Variable is not defined")
except ZeroDivisionError:
    print("Divided by zero")
finally:
    print("Finish exception handling")
```

(h)

```
def f():
    g()
def g():
    raise Exception("An error occurred")
try:
    f()
except Exception as e:
    print(e)
```

5. **[Function Basics & More]** In each of the following questions, you are asked to show what will be printed out. If there is an error, please explain why it is an error. (28%)

(a)

```
def reverse_string(s):
    if s == "":
        return s
    else:
        return reverse_string(s[1:]) + s[0]
print(reverse_string("Python"))
```

(b)

```
def power(base, exp):
    if exp == 0:
        return 1
    elif exp > 0:
        return base * power(base, exp - 1)
    else:
        return 1 / power(base, -exp)
print(power(2, -3))
```

(c)

```
def foo(a, b=2, c):
    return a + b + c
print(foo(1, 3))
```

(d)

```
def higher_order(func, *args):
    return func(*args)
print(higher_order(lambda x, y: x*y, 5, 2))
```

(e)

```
numbers = [1, 2, 3, 4, 5]
print(list(map(lambda x: x if x%2==0 else x**3, numbers)))
```

(f)

```
strings = ["apple", "banana", "cherry", "date"]
print(list(filter(lambda x: len(x) > 5, strings)))
```

(g)

```
def add_element(some_list):
    some_list.append(1)
my_list = [0]
add_element(my_list)
print(my_list)
```

(h)

```python
def change_value(x):
    x = x + 1
num = 0
change_value(num)
print(num)
```

(i)

```python
def power(base, exponent=2):
    return base ** exponent
print(power(2))
print(power(2, 3))
```

(j)

```python
def add_to(num, target=[]):
    target.append(num)
    return target
print(add_to(1))
print(add_to(2))
print(add_to(3))
```

(k)

```python
def apple():
    global x
    x = x + 1
def orange():
    return x + 1
def banana():
    x = 7
    return x + 1
x = 3
apple()
print(orange(), banana())
```

(l)

```python
x = 5
def apple():
    print(x)
x = 3
def orange(x):
    print(x)
def banana():
    x = 7
    print(x)
print(x)
apple()
orange(x)
print(x)
banana()
```

(m)

```
my_dict = {'a': (2, 1), 'b': (1, 2)}
sorted_dict = dict(sorted(my_dict.items(), key=lambda i: i[1]))
print(sorted_dict)
```

(n)

```
def binary_search(arr, low, high, x):
    if high >= low:
        mid = (high + low) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binary_search(arr, low, mid - 1, x)
        else:
            return binary_search(arr, mid + 1, high, x)
    else:
        return -1
result = binary_search([2, 3, 4, 10, 40], 0, 4, 10)
print(result)
```

6. Write a Python program to generate and print out the **Pascal's triangle** using *recursion*. Your program needs to be able to have the following **sample input/output.** (25%)

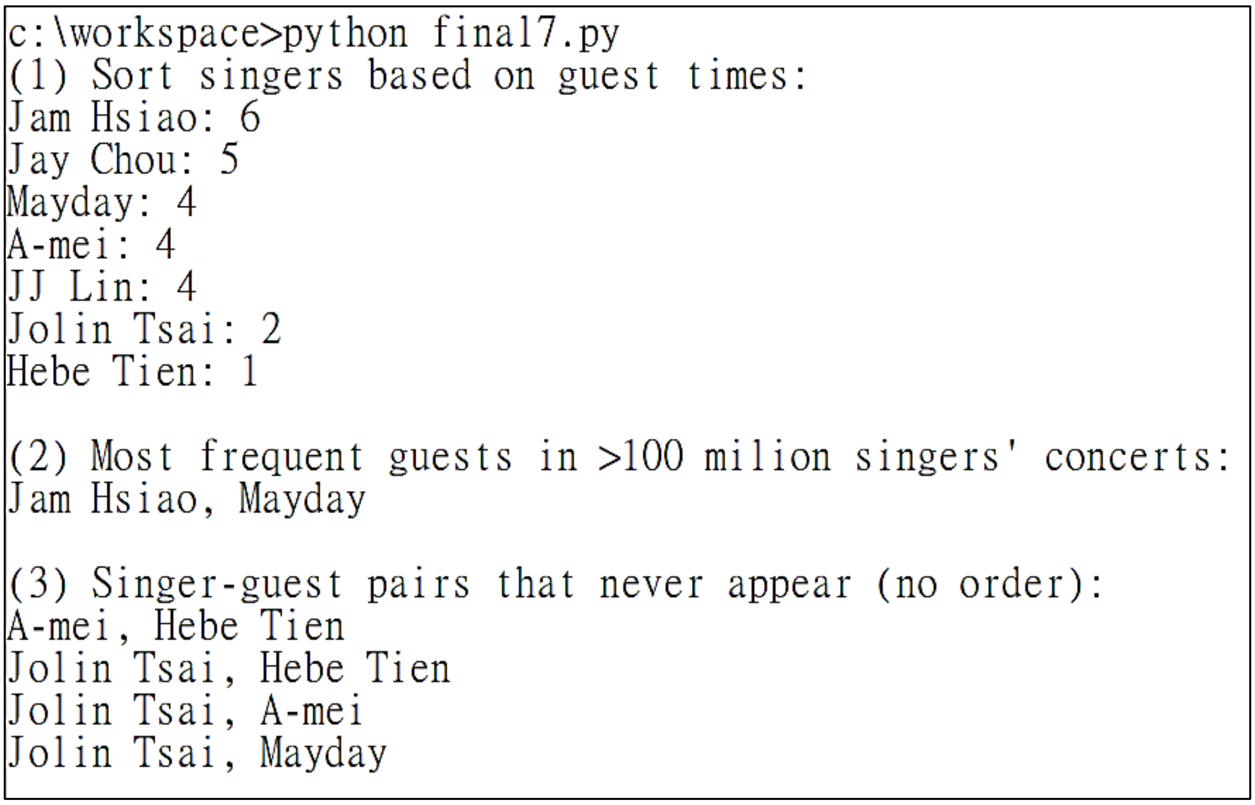
```
c:\workspace>python final2.py
Height of Pascal's triangle: 1
Direction of triangle ('normal' or 'reversed'): normal
1

c:\workspace>python final2.py
Height of Pascal's triangle: 5
Direction of triangle ('normal' or 'reversed'): normal
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1

c:\workspace>python final2.py
Height of Pascal's triangle: -1
Invalid input. Please enter an integer greater than or equal to 1.
Height of Pascal's triangle: 0
Invalid input. Please enter an integer greater than or equal to 1.
Height of Pascal's triangle: 8
Direction of triangle ('normal' or 'reversed'): xdxd
Invalid input for direction. Please enter 'normal' or 'reversed'.
Direction of triangle ('normal' or 'reversed'): reversed
1 7 21 35 35 21 7 1
  1 6 15 20 15 6 1
   1 5 10 10 5 1
     1 4 6 4 1
      1 3 3 1
       1 2 1
        1 1
         1

c:\workspace>
```

7. Given a data file, **taiwan_popular_singer.csv**, which shows some basic information of 7 Taiwan popular singers, including their names, popularity (number of fans in millions), names of guests who had ever appeared in their concerts, and the debut year. Your task is to write a program to answer the following three questions. It is important that you need to write a `function` to answer each question. In addition, your code is required to use `dictionary`. The sample output is shown in the following. (25%)

(1) Sort the singers based on the times that serve as guests in a descending order.
(2) Who are the singers that most frequently served as the guest in concerts whose singers' popularity is higher than 100 million?
(3) List the pairs of singers who never appear in each other's concerts.

taiwan_popular_singer.csv

```
singer,popularity(millions),guests,debut
Jay Chou,105,Jolin Tsai|Jam Hsiao|Mayday|A-mei,2000
Jam Hsiao,101,Mayday|A-mei|Jay Chou|JJ Lin,2007
JJ Lin,99,Jam Hsiao|Jolin Tsai|Jay Chou|A-mei,2003
Mayday,150,Jam Hsiao|A-mei|Hebe Tien|JJ Lin,1999
A-mei,135,Jay Chou|Jam Hsiao|Mayday,1996
Jolin Tsai,90,JJ Lin|Jam Hsiao|Jay Chou,1999
Hebe Tien,140,Jam Hsiao|Jay Chou|JJ Lin|Mayday,2001
```



```
c:\workspace>python final7.py
(1) Sort singers based on guest times:
Jam Hsiao: 6
Jay Chou: 5
Mayday: 4
A-mei: 4
JJ Lin: 4
Jolin Tsai: 2
Hebe Tien: 1

(2) Most frequent guests in >100 milion singers' concerts:
Jam Hsiao, Mayday

(3) Singer-guest pairs that never appear (no order):
A-mei, Hebe Tien
Jolin Tsai, Hebe Tien
Jolin Tsai, A-mei
Jolin Tsai, Mayday
```