1st Midterm Exam

2024.04.11 (Thursday) 10:10 – 12:00

1. **[Python Basics]** 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. (12%)

(a)	`print(9/5 - 9//5 + 9%5)`
(b)	`print(4 - 2 ** 2 + 12 / 2 ** 2)`

(c)	`m, x, b = 2, 3`
	`y = mx + b`
	`print(y)`

(d)	`alpha, x = 0.5, 10`
	`cofa()ctor = alpha * x * x`
	`_order = 1.0/cofa()ctor`
	`print(_order)`

(e)	`a, b = 3, 5`
	`a, b = b, a`
	`b, a == a, b`
	`print(a, b)`

(f)	`s = 3.14` `print(str(int(float(s)))`
(g)	`word = 'Coding'` `print(word[-1] + word[5] + word[len(word)-1])`

(h)	`s = 'pig'`
	`s[3] = 's'`
	`print(s)`

(i)	`print("Time left %03d:%05.3f" % (1, 12.34567))`
(j)	`s1, s2 = "ok", "go"` `print(2*s1 + "s2")`
(k)	`s = "oh_my_god!"` `print(s[:5])`
(l)	`s = "are_you_kidding_me?"` `print(s[5::-1])`

2. **[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. (18%)

(a)
```
time, money = 100, 0
print(not money and not time)
```

(b)
```
print(((2 or 0) + (0 and 8)) != 2)
```

(c)
```
cloudy = True
num = 0
print(cloudy or 5 / num > 3)
```

(d)
```
score = 21
while score >= 2:
    score = score/3 - 1
print(score)
```

(e)
```
x, y = 5, 0
while x <= y or y <= x:
    x, y = x - 1, y - 1
print("Done!")
```

(f)
```
a, b, i = 2, 4, 1
while i < 5:
    a, b, i = b - a, b + a, i + 1
print(a)
```

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

(h)
```
x, out = "6842915703", ""
i = 0
while i < 10:
    out += x.count(str(i)) * str(i)
    i += 1
print(out)
```

(i)
```
n = 20
while n != 1:
    print(n, end=", ")
    if n % 2 == 0:
        n = n // 2
    else:
        n = n * 3 + 1
print(n, end=".\n")
```

3. **[List]** 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. (26%)

(a)
```
d1, d2 = ["Taylor", "Swift", 2024], "2013"
d1[2], d2 = d2, d1[2]
print(d1, d2)
```

(b)
```
score = [90, 60, 30, 80]
print((min(score)+max(score)/len(score)))
```

(c)
```
game = ["rock", "paper"]
game = [game] + ["scissor"]
print(game)
```

(d)
```
n = [1, 3, 5, 2, 4, 6]
print(n[2:5] * 2)
```

(e)
```
notes = ["Do", "Re", "Mi"]
print(notes[2] == ["Mi"])
```

(f)
```
n = [1, 2]
n.append(3)
n.append([0])
n.extend(n)
print(n)
```

(g)
```
data = "Bulbasaur Charmander Pikachu"
print("->".join(data.split(" ") + ["Squirtle"]))
```

(h)
```
v = [1, 2, 3, 4]
p = v[:]
v[2] = 5
print(p)
print(v)
```

(i)
```
v = [1, 2, 3, 4]
p = v
v[2] = 5
print(p)
print(v)
```

(j)
```
x = [5, 3, 2, 6, 1]
x.remove(6)
x.pop()
x.insert(0, 0)
x.sort(reverse = True)
x.pop()
print(x)
```

(k)	`days = [['Mon','Oct',21], ['Wed','Nov',7]], ['Fri','Dec',6]]` `print(days[1][0], days[1:], days[0][0][1][1])`

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

(m)	`s = [1, 7, 4, 5, 3, 3]`
	`t = sorted(s)`
	`t.extend(8)`
	`print(t)`

4. Please read the following code (left block), and answer what will be printed out based on the respective initialized x in questions (a) – (d). (8%)

```
# initialize x here
if x % 2 != 0:
    if x ** 2 <= 36:
        print('pow')
    else:
        print(x // 3)
else:
    if x < 0:
        print(False)
    elif not x + 2 > 8:
        print(x / 2)
    else:
        print("end")
```

Questions	The initialized x	Printed?
(a)	x = 12	
(b)	x = 1	
(c)	x = 7	
(d)	x = -4	

5. Write a program that allows the user to input a time string in the format `'h:m:s'`, in which h is hour $(0 \le h \le 23)$, m is minute $(0 \le m \le 59)$, and s is second $(0 \le s \le 59)$. Then your program should convert time `'h:m:s'` to seconds in a day, and print the converted seconds. The sample input and output are shown in the following. (14%)

Example 1	`Enter a time (h:m:s): 1:10:25` `Seconds in a day = 4225`
Example 2	`Enter a time (h:m:s): 15:3:8` `Seconds in a day = 54188`

6. Please read the following code, and answer what will be printed out based on different initialized num1 and num2 in questions (a) and (b). (12%)

```
# initialize num1 and num2 here
ans = 0
m, n = len(num1), len(num2)
i = 0
while i < m:
    a = int(num1[m - 1 - i])
    j = 0
    while j < n:
        b = int(num2[n - 1 - j])
        ans += a * b * (10 ** (i + j))
        j += 1
    i += 1
print(ans)
```

Question	Initialized num1, num2	Printed?
(a)	`num1, num2 = "123", "321"`	
(b)	`num1, num2 = "999", "9"`	

7. Write a program to find *Perfect numbers* from 2 to n, where n is input by the user. According to Wikipedia : In the number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors. That said, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself). For example, the first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. Your program is required to accept an input as n, and output all of the perfect numbers from 2 to n in a list. An example is shown as below. (14%)

Example 1	**Example 2**
Input the range number: 1000 Perfect numbers: [6, 28, 496]	Input the range number: 10000 Perfect numbers: [6, 28, 496, 8128]

8. Please write a program with nested `while` loops (e.g., `while` in `while` in `while`) to generate and print a 9×9 multiplication table in a particular order, **as exactly shown in the right figure**. Note that if you totally use `print()` (i.e., without using while loops) to generate the multiplication table, you will get only 1%. (16%)

```
9 x 9 = 81    9 x 8 = 72    9 x 7 = 63
8 x 9 = 72    8 x 8 = 64    8 x 7 = 56
7 x 9 = 63    7 x 8 = 56    7 x 7 = 49
6 x 9 = 54    6 x 8 = 48    6 x 7 = 42
5 x 9 = 45    5 x 8 = 40    5 x 7 = 35
4 x 9 = 36    4 x 8 = 32    4 x 7 = 28
3 x 9 = 27    3 x 8 = 24    3 x 7 = 21
2 x 9 = 18    2 x 8 = 16    2 x 7 = 14
1 x 9 = 9     1 x 8 = 8     1 x 7 = 7

9 x 6 = 54    9 x 5 = 45    9 x 4 = 36
8 x 6 = 48    8 x 5 = 40    8 x 4 = 32
7 x 6 = 42    7 x 5 = 35    7 x 4 = 28
6 x 6 = 36    6 x 5 = 30    6 x 4 = 24
5 x 6 = 30    5 x 5 = 25    5 x 4 = 20
4 x 6 = 24    4 x 5 = 20    4 x 4 = 16
3 x 6 = 18    3 x 5 = 15    3 x 4 = 12
2 x 6 = 12    2 x 5 = 10    2 x 4 = 8
1 x 6 = 6     1 x 5 = 5     1 x 4 = 4

9 x 3 = 27    9 x 2 = 18    9 x 1 = 9
8 x 3 = 24    8 x 2 = 16    8 x 1 = 8
7 x 3 = 21    7 x 2 = 14    7 x 1 = 7
6 x 3 = 18    6 x 2 = 12    6 x 1 = 6
5 x 3 = 15    5 x 2 = 10    5 x 1 = 5
4 x 3 = 12    4 x 2 = 8     4 x 1 = 4
3 x 3 = 9     3 x 2 = 6     3 x 1 = 3
2 x 3 = 6     2 x 2 = 4     2 x 1 = 2
1 x 3 = 3     1 x 2 = 2     1 x 1 = 1
```