

NATIONAL UNIVERSITY OF SINGAPORE
Department of Computer Science, School of Computing
IT5001 – *Software Development Fundamentals*
Academic Year 2024/2025, Semester 2

Final Assessment
QUESTION BOOKLET

28 April 2025

Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES (please read carefully):

1. This is a **closed-book assessment**. You are allowed **ONE (1)** A4-sized reference sheet, double-sided, printed or written, and **ONE (1)** blank A4 paper for scratch.
2. You may use a non-programmable, NUS-approved calculator.
3. Use of any other electronic devices, including smart watches, is **NOT** allowed.
4. This paper comprises **FORTY-SEVEN (47) questions** and **TWENTY-FIVE (25) pages** including this cover page.
5. Do **NOT** write your name on any document you submit.
6. Write your **Student Number (starting with A)** on the 'ANSWER BOOKLET'.
7. Write and shade your **Student Number (starting with A)** on the 'OCR Answer Sheet' using a 2B pencil.
8. For Multiple-Choice Questions (MCQs) Q1 to Q38 (inclusive), write your answers in the empty space beside the question.
9. For the remaining questions, write your answers in the space provided; no extra sheets will be accepted as answers. You may write in blue or black with a pencil or pen.
10. The total attainable score for this assessment is **100 marks**. You must complete all questions to score full marks. This assessment counts towards **40%** of your final grade.
11. You **cannot** communicate with anyone other than the invigilators throughout the exam.

You **must attempt the assessment on your own**. The University takes a zero-tolerance approach towards plagiarism and cheating.

MULTIPLE CHOICE QUESTIONS [50 marks]

For each question, choose only **ONE (1)** answer.

1. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
False or (False and True) or not False
```

- A. True
- B. False
- C. None
- D. 0
- E. Evaluating this expression yields an error

2. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
1 + 2 % 0 + -3
```

- A. 1
- B. 0
- C. -1
- D. -2
- E. Evaluating this expression yields an error

3. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
int(int(1/3 + 5//3)**2 + float(2/3)) or False
```

- A. True
- B. 4
- C. 1
- D. 2.0
- E. Evaluating this expression yields an error

4. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
'sesquipedalian'[8]
```

- A. 'a'
- B. 'd'
- C. 'e'
- D. 'p'
- E. Evaluating this expression yields an error

5. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
'le'[1] + 'poisson'[3] + 'steve'[1:]
```

- A. 'lsteve'
- B. 'listeve'
- C. 'lssteve'
- D. 'est'
- E. 'esteve'

6. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
'May the 4th be with you'[-5::-2][:2] * 2
```

- A. 'hihi'
- B. 'hyhy'
- C. 'iyiy'
- D. '' # Empty string
- E. Evaluating this expression yields an error

7. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
['alice', 'betty', 'carol'][2][-1]
```

- A. 'c'
- B. 'e'
- C. 'y'
- D. 'l'
- E. Evaluating this expression yields an error

8. What is the output of the following code? **[1 mark]**

```
lst = list({x: x+1 for x in range(4)})  
print(lst[2])
```

- A. 3
- B. 2
- C. 1
- D. 0
- E. Executing this code yields an error

9. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
('A', ('B',)), ('C', 1), 'D')[2]
```

- A. 'B'
- B. 'C'
- C. ('B',)
- D. ('C', 1)
- E. Evaluating this expression yields an error

10. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
tuple('cold'[1]) + tuple('range')
```

- A. ('l', 'range')
- B. ('o', 'range')
- C. ('l', 'r', 'a', 'n', 'g', 'e')
- D. ('o', 'r', 'a', 'n', 'g', 'e')
- E. Evaluating this expression yields an error

11. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
set('leonardo') | set('donatello')
```

- A. {'r'}
- B. {'r', 't'}
- C. {'a', 'd', 'e', 'l', 'n', 'o'}
- D. {'a', 'd', 'e', 'l', 'n', 'o', 'r', 't'}
- E. Evaluating this expression yields an error

12. What is the output of the following code? **[1 mark]**

```
side_a = {0 & 1 | 2}  
side_b = {(3,4), 5, (6,7)}  
res = side_b - side_a  
print(res)
```

- A. set() # empty set
- B. {2}
- C. {0, 2}
- D. {(3, 4), 5, (6, 7)}
- E. Executing this code yields an error

13. What is the output of the following code? **[1 mark]**

```
public_keys = {'alice': [2, 4, 5],
               'bob': [1, 8, 7],
               'darth': [6, 3, 9]}
print(public_keys.get('bob', [0]))
```

- A. [0]
- B. [1, 8, 7]
- C. [2, 4, 5]
- D. [6, 3, 9]
- E. Executing this code yields an error

14. Given the following definition of my_dict:

```
my_dict = {'object': 'color',
           'citrus': 'orange',
           'owl': 'brown',
           'paper': 'white'}
```

If using my_dict is a must, which of the following will produce output 'citrus'? **[1 mark]**

- A. list(my_dict.keys())[1]
- B. my_dict.get('citrus')
- C. [my_dict.keys()][1]
- D. All of the above
- E. None of the above

15. What is the output of the following code? [1 mark]

```
school = {
    'my_class': {
        'george': {'math': 90, 'english': 42},
        'harold': {'math': 75, 'english': 80},
    }
}

hack = school['my_class']
hack['melvin'] = {'math': 100, 'english': 100}
hack['george']['english'] = 12

school['my_class']['melvin'] = 0
print(school['my_class'])
```

- A. {'george': {'math': 90, 'english': 12},
 'harold': {'math': 75, 'english': 80},
 'melvin': 0}
- B. {'george': {'math': 90, 'english': 42},
 'harold': {'math': 75, 'english': 80},
 'melvin': 0}
- C. {'george': {'math': 90, 'english': 12},
 'harold': {'math': 75, 'english': 80}}
- D. Executing this code yields an error
- E. None of the above

16. What is the output of the following code? [1 mark]

```
def bar(x):
    if x <= 10:
        return x
    elif x % 2 == 0:
        return bar(x-1)
    else:
        return bar(x // 3)

print(bar(79))
```

- A. 9
- B. 8
- C. 7
- D. 6
- E. 5

17. What is the output of the following code? [1 mark]

```
def get_relative_dir(src, dest):
    x, y = '', ''
    pos, neg = 'EN', 'WS'

    if dest[0]-src[0] > 0:
        x = pos[0]
    elif dest[0]-src[0] < 0:
        x = neg[0]

    if dest[1]-src[1] > 0:
        y = pos[1]
    elif dest[1]-src[1] < 0:
        y = neg[1]

    return y + x

print(get_relative_dir((1, 3), (-1, 4)))
```

- A. NE
- B. NW
- C. W
- D. S
- E. SE

18. Observe the following code:

```
target = 3

def foo(x):
    if x == target:
        return ''
    if x % 2 == 1:
        return 'A' + foo(x+1)
    else:
        return 'A' + bar(x)

def bar(x):
    if x % 2 == 0:
        return 'B' + foo(x // 2)
    return foo(x//2)
```

Executing `foo(20)` produces a string. How long is this string? [1 mark]

- A. 7 characters long
- B. 6 characters long
- C. 5 characters long
- D. All of the above
- E. None of the above

19. What is the output of the following code? [1 mark]

```
x, i = '', 3
while i < 8:
    x += str(i)
    i += 1

print(x)
```

- A. 34567
- B. 345678
- C. 4567
- D. 45678
- E. Executing this code yields an error

20. What is the output of the following code? [1 mark]

```
def flat(seq):
    res = tuple()
    for item in seq:
        if type(item) != int:
            res = res + flat(item)
        else:
            res = res + (item,)
    return res

print(flat((7, [8, 9], (6, 5), 4, (1, [2, 3]))))
```

- A. (7, 8, 9, 6, 5, 4, 1, 2, 3)
- B. [7, 8, 9, 6, 5, 4, 1, 2, 3]
- C. (7, 8, 9, 6, 5, 4, 1, [2, 3])
- D. [7, 8, 9, 6, 5, 4, 1, [2, 3]]
- E. Executing this code yields an error

21. Observe the following code:

```
def foo(f, lst):
    def quz(lst):
        return ''.join(lst)

    return lst if not lst else quz(f(lst))

def bar(lst):
    def baz(lst):
        s1 = [item for item in lst if item.isalpha()]
        s2 = [item for item in lst if item.isnumeric()]
        return s1 + s2

    return baz(lst)
```


Which of the following statements about the output produced by `foo(bar, 'PA55WORD')` is TRUE? [1 mark]

- A. An empty string is produced
- B. A string with alphabets trailing after numeric values is produced
- C. A string with numeric values trailing after alphabets is produced
- D. A string is produced by removing the alphabet values in input string 'PA55WORD'
- E. A string is produced by removing the numeric values in input string 'PA55WORD'

22. Given that `n` is a positive integer, what does the following function return? [1 mark]

```
def shazam(n):  
    if n < 0:  
        return []  
    return shazam(n-2) + [n]
```

- A. A list of integers starting from 0 or 1 to `n` (inclusive), increasing by 2
- B. A list of integers starting from 0 or 1 to `n` (exclusive), increasing by 2
- C. A list of integers starting from `n` (inclusive) to 0 or 1, decreasing by 2
- D. A list of integers starting from `n` (exclusive) to 0 or 1, decreasing by 2
- E. The function throws an error

23. What is the output of the following code? [1 mark]

```
def inside_out(seq):  
    if not seq:  
        return seq  
  
    if type(seq) == list:  
        if len(seq[-1]) < 3:  
            return [seq[-1]] + inside_out(seq[:-1])  
        return [inside_out(seq[-1])] + inside_out(seq[:-1])  
  
    return seq[-1] + inside_out(seq[:-1])  
  
msg = ['that', 'is', 'a', 'covert', 'message']  
print(inside_out(msg))
```

- A. ['taht', 'is', 'a', 'trevoc', 'egassem']
- B. ['taht', 'si', 'a', 'trevoc', 'egassem']
- C. ['egassem', 'trevoc', 'a', 'is', 'taht']
- D. ['egassem', 'trevoc', 'a', 'si', 'taht']
- E. Executing this code yields an error

24. The following Python expression is entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the following expression. **[1 mark]**

```
list(filter(lambda x: x<50, [1,4,9,16,25,36,49,64,81,100]))
```

- A. [1, 4, 9, 16, 25, 36, 49, 64]
- B. [1, 4, 9, 16, 25, 36, 49]
- C. [81, 100]
- D. [64, 81, 100]
- E. [49, 64, 81, 100]

25. What is the output of the following code? **[1 mark]**

```
from functools import reduce

def f1(x, y, z):
    return x + y*2 + z + y*2

f2 = lambda char: {'a': 2, 'w': 5, 'y': 1}.get(char, 0)
f3 = lambda x, y: x + y

temp = f1('w', 'y', 'a') * 2
print(reduce(f3, (map(f2, temp))))
```

- A. 0
- B. 11
- C. 22
- D. 44
- E. Executing this code yields an error

26. What is the output of the following code? **[1 mark]**

```
class Book:
    def __init__(self, title, author, num_pages):
        self.title = title
        self.author = author
        self.num_pages = num_pages
        self.info = f'{title} ({author}): {num_pages} pages'

book1 = Book('Open When', 'Julie', 300)
print(book1.info)
```

- A. {Open When} ({Julie}): {300} pages
- B. {Open When} {Julie}: {300} pages
- C. Open When (Julie): 300 pages
- D. Open When Julie: 300 pages
- E. Executing this code yields an error

27. What is the output of the following code? [1 mark]

```
class Rectangle:
    def __init__(self, width, height):
        self.width = max(1, width)
        self.height = max(1, height)

    def area(self):
        return self.width * self.height

    def compare(self, other_rect):
        if self.area() < other_rect.area():
            return "Smaller"
        if self.area() > other_rect.area():
            return "Larger"
        return "Same"

test_rect = Rectangle(16, 4)
for w, h in [(4, 0), (18, 6), (8, 8)]:
    print(test_rect.compare(Rectangle(w, h)))
```

- A. Larger
Smaller
Same
- B. Smaller
Larger
Same
- C. Larger
Same
Smaller
- D. Smaller
Same
Larger

E. Executing this code yields an error

28. What is the output of the following code? [1 mark]

```
class EqualShape:
    def __init__(self, width):
        self.width = max(1, width)

class Triangle(EqualShape):
    def __init__(self, width, height):
        super().__init__(width)
        self.height = height

    def get_area(self):
        return self.width * self.height // 2

class Square(EqualShape):
    def get_area(self):
        return self.width ** 2

t, r = Triangle(3, 3), Square(4)
print(t.get_area() + r.get_area())
```

- A. 20
- B. 20.0
- C. 20.5
- D. 40
- E. Executing this code yields an error

29. What is the output of the following code? [1 mark]

```
class M1():
    def __init__(self):
        print("Mamat")

class M2(M1):
    def act(self):
        print("Minah")

class M3(M2):
    def act(self):
        super().act()
        print("Joyah")

(M3()).act()
```

- A. Mamat
Minah
Joyah
- B. Minah
Joyah
- C. Joyah
- D. Mamat
Joyah
- E. Executing this code yields an error

30. Broadcasting is a feature originating from... [1 mark]

- A. imageio
- B. numpy
- C. PIL
- D. PILLOW
- E. None of the above

31. You can extract the exif data with the package... [1 mark]

- A. imageio
- B. math
- C. numpy
- D. PIL
- E. None of the above

32. How can you install extra Python packages from the internet? **[1 mark]**

- A. import
- B. PIL
- C. pip
- D. sys
- E. You cannot install any more packages

33. What is the output of the following code? **[3 marks]**

```
def f(x):
    if x < 10:
        return x
    if x % 2 == 1:
        return g(x-3)
    return f(x//2)

def g(x):
    if x < 10:
        return x
    if x % 3 == 2:
        return f(x-4)
    return g(x//2)

print(f(100))
```

- A. 1
- B. 4
- C. 5
- D. 7
- E. Executing this code yields an error

34. What is the output of the following code? **[3 marks]**

```
data = {}
for k in range(6,9):
    v = k**2
    while not 0 < v < 10:
        v = sum(int(x) for x in str(v))

    data[k] = v

print(data)
```

- A. {6: 9, 7: 4, 8: 1}
- B. {6: 9, 7: 4, 8: 1, 9: 9}
- C. {6: 9, 7: 13, 8: 10}
- D. {6: 9, 7: 13, 8: 10, 9: 9}
- E. Executing this code yields an error

35. What is the output of the following code? [3 marks]

```
lst = [27, 42, 24, 16, 9, 17, 12, 23, 32, 3]

mid_index = len(lst) // 2
left, right = lst[:mid_index], lst[mid_index:]

res = []
while left and right:
    if left[-1] < right[-1]:
        res.append(left.pop())
    else:
        res.append(right.pop())

res.extend(left)
res.extend(right)
print(res)
```

- A. [3, 9, 12, 16, 17, 23, 24, 27, 32, 42]
- B. [3, 9, 16, 24, 32, 23, 12, 17, 42, 27]
- C. [17, 12, 23, 27, 32, 3, 42, 24, 16, 9]
- D. [42, 32, 27, 24, 23, 17, 16, 12, 9, 3]
- E. None of the above

36. What is the output of the following code? [3 marks]

```
interchanges = {
    "chinatown": ["bayfront", "dhoby_gaut", "outram_park",
"tampines"],
    "dhoby_gaut": ["chinatown", "city_hall", "little_india",
"orchard", "promenade"],
    "harbourfront": ["buona_vista", "outram_park"],
    "little_india": ["bugis", "chinatown", "newton", "serangoon"],
    "outram_park": ["buona_vista", "chinatown", "harbourfront",
"marina_bay", "orchard", "raffles_place"],
    "serangoon": ["bishan", "little_india", "macpherson"],
}

def third_stops(start):
    if start not in interchanges:
        return start
    to_stop = []
    for transit in interchanges[start]:
        if transit not in interchanges:
            to_stop.append(transit)
    return to_stop

print(third_stops("little_india"))
```

- A. ['buona_vista']
- B. ['bayfront', 'tampines']
- C. ['bishan', 'macpherson']
- D. ['bugis', 'newton']
- E. ['city_hall', 'orchard', 'promenade']

37. What is the output of the following code? **[3 marks]**

```
def deep_reverse(seq):
    if type(seq) != list or not seq:
        return seq

    return deep_reverse(seq[1:]) + [deep_reverse(seq[0])]

l = [1, 2, [3, 4, [5, 6, [7], 8], 9], 10, [11]]
print(deep_reverse(l))
```

- A. [11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
- B. [11, 10, [9, 8, [7, 6, [5], 4], 3], 2, [1]]
- C. [[11], 10, [9, [8, [7], 6, 5], 4, 3], 2, 1]
- D. [[1], 2, [3, [4, [5], 6, 7], 8, 9], 10, 11]
- E. Executing this code yields an error

38. What is the output of the following code? **[3 marks]**

```
def f(seq):
    is_vowel = lambda c: c in 'aeiou'

    if not seq:
        return seq
    if type(seq) == str:
        return [seq, ' '.join(filter(is_vowel, seq))]

    return f(seq[0]) + f(seq[1:])

lst = ['alan', ['code', 'pray', 'love']]
print(f(lst))
```

- A. ['alan', 'a', 'code', 'oe', 'pray', 'a', 'love', 'oe']
- B. ['alan', 'aa', 'code', 'oe', 'pray', 'a', 'love', 'oe']
- C. ['alan', 'ln', 'code', 'cd', 'pray', 'pry', 'love', 'lv']
- D. [['alan', 'aa'], ['code', 'oe'], ['pray', 'a'], ['love', 'oe']]
- E. [['alan', 'ln'], ['code', 'cd'], ['pray', 'pry'], ['love', 'lv']]

FILL-IN-THE-BLANKS QUESTIONS [48 marks]

For each question, replace each blank with a valid Python expression/statement. Note that your code needs to be syntactically correct to gain marks. You cannot use any semicolon (;), and we will deduct marks if your answer is too long.

39. For a word with n characters, we want to print it n times. Write a function `recite(s)` to do this job.

Sample output:

```
>>> recite('apple')
apple
apple
apple
apple
apple
apple
apple
>>> recite('boy')
boy
boy
boy
```

Complete the function by filling in the blank(s). [6 marks]

```
def recite(s):
    for i in ____BLANK_1____:
        print(s)
```

BLANK	ANSWER
1	

40. The latest research from Stupendous University shows that a marriage or partnership will be successful if the ages of the two persons:
- differ by less than or equal to 5
 - are not the same

Write a function `match(a1, a2)` which returns `True` if a couple with the two ages `a1` and `a2` match **ALL** the above criteria, and `False` otherwise.

Sample output:

```
>>> print(match(25,20))
True
>>> print(match(25,19))
False
>>> print(match(25,25))
False
```

Complete the function by filling in the blank(s). [6 marks]

```
def match(a1, a2):  
    return ____BLANK_1____
```

BLANK	ANSWER
1	

41. Singapore NRIC Validation

As we mentioned in class, Singapore NRIC (National Registration Identity Card) has a checksum mechanism to check if the IC is valid. The structure of the NRIC number/FIN is

@xxxxxxxx#,

where @ is either one of letters "S", "T", "F", "G" or "M", depending on the status of the holder.

Here are some of the IC numbers for some famous people in Singapore:

- S0000001I (Yusof bin Ishak, first President of Singapore)
- S0000002G (Wee Chong Jin, first Chief Justice of Singapore)
- S0000003E (Lee Kuan Yew, first Prime Minister of Singapore)
- S0000004C (Kwa Geok Choo, wife of Lee Kuan Yew)
- S0000005A (Toh Chin Chye, first Deputy Prime Minister of Singapore)
- S0000006Z (Goh Keng Swee, second Deputy Prime Minister of Singapore)
- S0000007H (S Rajaratnam, first Minister of Culture)

Finally, symbol # represents a checksum value. The steps involved to compute the final # checksum are as follows:

1. Multiply each digit in the NRIC number by its weight (i.e., 2 7 6 5 4 3 2, in order).
2. Add together the above products.
3. If the first letter (i.e., UIN of the NRIC) starts with T or G, add 4 to the total.
4. Find the remainder of (sum calculated above) mod 11.
5. If the NRIC starts with F or G: 0=X, 1=W, 2=U, 3=T, 4=R, 5=Q, 6=P, 7=N, 8=M, 9=L, 10=K
6. If the NRIC starts with S or T: 0=J, 1=Z, 2=I, 3=H, 4=G, 5=F, 6=E, 7=D, 8=C, 9=B, 10=A

Your job is to complete the function `last_letter(s)` that takes in the first letter and the 7 digits of an NRIC number (@xxxxxxxx) and return the last letter (#).

Sample output:

```
>>> print(last_letter('S0000003'))  
E  
>>> print(last_letter('S1234567'))  
D  
>>> print(last_letter('S8888888'))  
Z
```

Complete the function by filling in the blank(s). [6 marks]

```
def last_letter(s):
    total = 0
    w = [2,7,6,5,4,3,2]
    FG = 'XWUTRQPNMLK'
    ST = 'JZIHGFEDCBA'
    for i in ____BLANK_1____:
        total += ____BLANK_2____
    if ____BLANK_3____:
        total += 4
    r = ____BLANK_4____ # calculating the remainder
    if s[0] in ['F', 'G']:
        return FG[r]
    else:
        return ST[r]
```

BLANK	ANSWER
1	
2	
3	
4	

42. You are given a list of sorted integers. Write a tail-recursive function `rd(lst)` that returns a new list with all duplicate elements removed while preserving the order of the first occurrence of each element. ("rd" stands for "remove duplicates")

You must implement the core logic using tail recursion (i.e., the recursive call must be in tail position), and avoid using `set()`, loops or list comprehension for the main logic.

Sample output:

```
>>> rd([1,2,3,4,5,6,7,8])
[1, 2, 3, 4, 5, 6, 7, 8]
>>> rd([1, 3, 3, 4, 5, 8, 8, 8])
[1, 3, 4, 5, 8]
>>> rd([1])
[1]
>>> rd([4,4,4,4])
[4]
>>> rd([])
[]
```

Complete the function by filling in the blank(s). [6 marks]

```
def rd(lst):
    if ____BLANK_1____:
        return lst
    if ____BLANK_2____:
        return ____BLANK_3____
    else:
        return ____BLANK_4____
```

BLANK	ANSWER
1	
2	
3	
4	

43. Given a deep list that contains only integers or lists similar to our definitions in class, write a function `deep_sum(lst)` to compute the sum of all the numbers in the deep list.

Sample output:

```
>>> print(deep_sum([1,2,[3],4,5,6,7],[8,9,10]))
55
>>> print(deep_sum([1,2,[3,4,5,[6],[7,8],[9,[10]]]]))
55
```

Complete the function by filling in the blank(s). [6 marks]

```
def deep_sum(seq):
    if seq == []:
        return ____BLANK_1____
    elif ____BLANK_2____:
        return ____BLANK_3____
    else:
        return ____BLANK_4____
```

BLANK	ANSWER
1	
2	
3	
4	

44. You want to do an inventory check with your stock and your assistant is shouting out all the items one by one in order of a list like below:

```
['orange', 'apple', 'apple', 'mango', 'orange', 'apple']
```

Write a function `inventory(lst)` that converts an inventory list into a dictionary form. The keys are the items' names and the values are the number of times the corresponding item appears in the list.

Sample output:

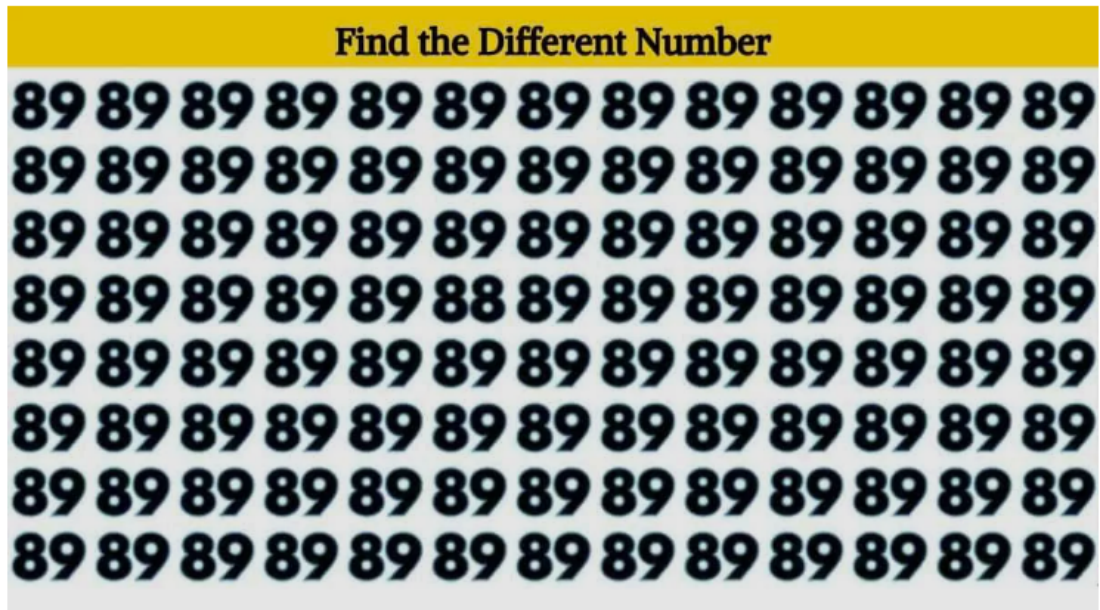
```
>>>
print(inventory(['orange','apple','apple','mango','orange','apple'
']))
{'orange': 2, 'apple': 3, 'mango': 1}
>>> print(inventory(['pen','pencil','pen','eraser','pen']))
{'pen': 3, 'pencil': 1, 'eraser': 1}
```

Complete the function by filling in the blank(s). [6 marks]

```
def inventory(lst):
    inv = {}
    for ____BLANK_1____:
        if ____BLANK_2____:
            ____BLANK_3____
            inv[item] += 1
    return inv
```

BLANK	ANSWER
1	
2	
3	

45. On social media, some people will post "puzzles", just like the following where you are supposed to look for the number:



Here, all but one of the numbers is "89". There is one instance of "88" in the middle.

Given a 2D array defined during lecture, your job is to complete the function `find_two_numbers(m)` which returns the two distinct numbers found in the 2D array. You can assume that you are given a 2D array that only contains two different integers.

You must implement the core logic using loops and avoid using `set()` or recursion for the main logic.

Sample output:

```
>>> print(find_two_numbers([[1,1,1],[1,1,2],[1,1,1],[1,1,1]]))
[1, 2]
>>>
print(find_two_numbers([[89,89,89,89],[89,89,88,89],[89,89,89,89]
]))
[88, 89]
```

Complete the function by filling in the blank(s). [6 marks]

```
def find_two_numbers(m):
    output = []
    r = len(m)
    c = len(m[0])
    for ____BLANK_1____ in ____BLANK_2____:
        for ____BLANK_3____ in ____BLANK_4____:
            if ____BLANK_5____:
                output.append(____BLANK_6____)
    return sorted(output)
```

BLANK	ANSWER
1	
2	
3	
4	
5	
6	

46. The Quest to Save the King

In a distant land, there existed a mysterious kingdom governed by 3 omnipotent forces – The Power of Doubling, The Power of Tripling, and The Power of Subtraction. The King of this land abused these powers during his youth, later in life causing him to be cursed with a magical number n — a number so large that it caused everything within 6 feet from him to age n times faster.

Desperate to be embraced by the Queen's hugs again without ending her life abruptly, the King sought to break the curse by reducing n to 1. But there is a catch: the King can only use one of the three forces at a time by doing the following:

- Divide by 2 if n is divisible by 2.
- Divide by 3 if n is divisible by 3.
- Subtract 1 from n (must if neither division is possible).

The King has no idea how many steps it will take to reach the magical number 1. He is starting from a large number n , and with each step, he must decide which power to use.

Write a function `min_steps(n)` to determine the minimum number of steps required for the King to reduce n to 1, by choosing the best move at each step.

For instance, if the King starts with $n = 10$, here is one possible sequence of steps:

- Start at 10: Subtract 1 $\rightarrow 9$
- At 9 (divisible by 3): Divide by 3 $\rightarrow 3$
- At 3 (divisible by 3): Divide by 3 $\rightarrow 1$

In this scenario, it took 3 steps for the King to reach his goal.

Complete the function by filling in the blank(s). **[6 marks]**

```
def min_steps(n):  
    if n == 1:  
        return 0  
  
    steps = ____BLANK_1____  
  
    if ____BLANK_2____:  
        steps = ____BLANK_3____  
    if ____BLANK_4____:  
        steps = ____BLANK_5____  
  
    return ____BLANK_6____
```

BLANK	ANSWER
1	
2	
3	
4	
5	
6	

FREE OPINION QUESTION [2 marks]

47. <redacted>

END OF QUESTION PAPER

NATIONAL UNIVERSITY OF SINGAPORE
Department of Computer Science, School of Computing
IT5001 – *Software Development Fundamentals*
Academic Year 2024/2025, Semester 2

Final Assessment
SOLUTIONS MANUAL

28 April 2025

Time allowed: 2 hours

MULTIPLE CHOICE QUESTIONS [50 marks]

1. A	2. E	3. C	4. B	5. E	6. A	7. D
8. B	9. D	10. D	11. D	12. D	13. B	14. A
15. A	16. B	17. B	18. A	19. A	20. A	21. C
22. A	23. C	24. B	25. C	26. C	27. A	28. A
29. A	30. B	31. D	32. C	33. D	34. A	35. B
36. D	37. C	38. B				

FILL-IN-THE-BLANKS QUESTIONS [48 marks]

NOTE: The following are only ideas for solving these questions; better solutions exist, and if your code follows a different approach but still returns the correct output (and within 50 characters or less per blank), you may consider your solution correct.

39.	<pre>def recite(s): for i in range(len(s)): # BLANK 1 print(s)</pre>
40.	<pre>def match(a1,a2): return 0 < abs(a1-a2) <= 5 # BLANK 1</pre>

41.	<pre> def last_letter(s): total = 0 w = [2,7,6,5,4,3,2] FG = 'XWUTRQPNMLK' ST = 'JZIHGFEDCBA' for i in range(1,8): # BLANK 1 total += w[i-1]*int(s[i]) # BLANK 2 if s[0] in ['T', 'G']: # BLANK 3 total += 4 r = total % 11 # BLANK 4 if s[0] in ['F', 'G']: return FG[r] else: return ST[r] </pre>
42.	<pre> def rd(lst): if len(lst) < 2: # BLANK 1 return lst if lst[0] == lst[1]: # BLANK 2 return rd(lst[1:]) # BLANK 3 else: return [lst[0]] + rd(lst[1:]) # BLANK 4 </pre>
43.	<pre> def deep_sum(seq): if seq == []: return 0 # BLANK 1 elif type(seq) != list: # BLANK 2 return seq # BLANK 3 else: output = deep_sum(seq[0]) + deep_sum(seq[1:]) # BLANK 4 </pre>
44.	<pre> def inventory(lst): inv = {} for item in lst: # BLANK 1 if not item in inv: # BLANK 2 inv[item] = 0 # BLANK 3 inv[item] += 1 return inv </pre>

45.	<pre> def find_two_numbers(m): output = [] r = len(m) c = len(m[0]) for row in m: for x in row: if not x in output: output.append(x) return sorted(output) </pre>
46.	<pre> def min_steps(n): <i># Base case: If the King has reached 1, no further steps are needed.</i> if n == 1: return 0 <i># Step 1: Subtract 1 and recurse to the next step</i> steps = min_steps(n - 1) <i># Step 2: If the King can divide by 2, try dividing and recurse</i> if n % 2 == 0: steps = min(steps, min_steps(n//2)) <i># Step 3: If the King can divide by 3, try dividing and recurse</i> if n % 3 == 0: steps = min(steps, min_steps(n//3)) <i># Return 1 step + the minimum result of all possibilities</i> return 1 + steps </pre>

END OF SOLUTIONS MANUAL