Queue

Standard difficulty · 20 questions · 40 marks · Time: 40 minutes

Name: _____ Date: _____ Score: ____ / 40

Show your working for Sections B and C. Answers are on the last page — try every question first!

Section A — Choose the correct answer 8 × 1 = 8 marks

1. A queue works on the principle of:

(a) largest first **(b)** random order

(c) Last In, First Out (LIFO) **(d)** First In, First Out (FIFO)

2. What is the last line this program prints?

```
def enqueue(q, x):
    q.append(x)

def dequeue(q):
    if len(q) == 0:
        return 'Underflow'
    return q.pop(0)

q = []
enqueue(q, 86)
enqueue(q, 89)
enqueue(q, 81)
print(dequeue(q))
enqueue(q, 63)
print(dequeue(q))
enqueue(q, 83)
print(q)
```

(a) [81, 63, 83] **(b)** [63, 81, 83] **(c)** [63, 83] **(d)** [81, 83, 63]

3. What is the first line this program prints?

```
def enqueue(q, x):
    q.append(x)

def dequeue(q):
    if len(q) == 0:
        return 'Underflow'
    return q.pop(0)

q = []
enqueue(q, 55)
enqueue(q, 58)
enqueue(q, 75)
print(dequeue(q))
print(dequeue(q))
enqueue(q, 91)
print(dequeue(q))
enqueue(q, 19)
print(q)
```

(a) 55 **(b)** 19 **(c)** None **(d)** 58

4. What does this print?

```
d = [17, 37]
d.pop(0)
d.insert(0, 94)
d.pop()
d.insert(0, 97)
d.pop(0)
d.append(97)
print(d)
```

(a) [94, 97, 17] **(b)** [97, 94] **(c)** [94, 97, 37] **(d)** [94, 97]

5. Which data structure suits plates piled one on another?

(a) list sorted by size **(b)** queue

(c) deque **(d)** stack

6. 42, 81, 61, 95, 49, 80, 92, 74, 71, 65, 35, 47 are enqueued in this order into an empty queue, and then 10 dequeue operations are done. Which value is now at the front?

(a) 65 **(b)** 61 **(c)** 35 **(d)** 47

7. What does this print?

```
d = list('civic')
ok = True
while len(d) > 1:
    if d.pop(0) != d.pop():
        ok = False
        break
print(ok)
```

(a) False **(b)** an IndexError **(c)** True **(d)** None

8. An empty queue gets 9 enqueue operations, then 8 dequeue operations, then 3 more enqueue operations. How many items are in the queue?

(a) 20 **(b)** 4 **(c)** 1 **(d)** 5

Section B — Short answer — 8 × 2 = 16 marks

9. What is a queue? Give an example from daily life.

10. What does this program print? Write each line in order, separated by commas.

```
def enqueue(q, x):
    q.append(x)

def dequeue(q):
    if len(q) == 0:
        return 'Underflow'
    return q.pop(0)

q = []
enqueue(q, 28)
enqueue(q, 67)
print(dequeue(q))
print(dequeue(q))
print(dequeue(q))
enqueue(q, 11)
print(dequeue(q))
enqueue(q, 21)
enqueue(q, 10)
print(q)
```

11. What does this print?

```
d = [47, 46]
d.insert(0, 95)
d.insert(0, 91)
d.insert(0, 99)
d.pop()
d.append(87)
print(d)
```

12. What does this print?

```
d = list('stats')
ok = True
while len(d) > 1:
    if d.pop(0) != d.pop():
        ok = False
        break
print(ok)
```

13. Write the values removed from the queue, in order, separated by commas.

```
def enqueue(q, x):
    q.append(x)

def dequeue(q):
    if len(q) == 0:
        return 'Underflow'
    return q.pop(0)

q = []
enqueue(q, 13)
print(dequeue(q))
enqueue(q, 75)
enqueue(q, 70)
print(dequeue(q))
enqueue(q, 51)
print(q)
```

14. Write the values removed from the deque, in the order they are removed, separated by commas.

```
d = [47, 43]
d.pop(0)
d.insert(0, 90)
d.insert(0, 65)
d.pop(0)
print(d)
```

15. 35, 92, 32, 15, 79 are enqueued in this order into an empty queue. How many dequeue operations are needed before 92 is at the front?

16. Which data structure (stack, queue or deque) suits plates piled one on another?

Section C — Word problems — 4 × 4 = 16 marks

17. Write a menu-driven program to implement a queue using a list, with enqueue, dequeue, peek and display.

18. Trace this program. (a) Write what each print(dequeue(q)) prints, in order. (b) What is the final queue? (c) Which value is at the front at the end?

```
def enqueue(q, x):
    q.append(x)

def dequeue(q):
    if len(q) == 0:
        return 'Underflow'
    return q.pop(0)

q = []
enqueue(q, 71)
enqueue(q, 96)
print(dequeue(q))
enqueue(q, 12)
print(dequeue(q))
enqueue(q, 26)
enqueue(q, 92)
print(q)
```

19. (a) Write the deque after each operation on it, in order. (b) What does it print? (c) Which two operations make this list behave as a stack?

```
d = [16, 23]
d.append(66)
d.pop(0)
d.pop()
d.insert(0, 95)
print(d)
```

20. (a) What does this print? (b) Explain how the program uses the list as a deque. (c) What would it print for the word noon?

```
d = list('stats')
ok = True
while len(d) > 1:
    if d.pop(0) != d.pop():
        ok = False
        break
print(ok)
```

Want another sheet on this chapter? In the Ambiful Worksheets app every sheet is new, with its answer key, and your child never gets the same question twice.

Scan the code to get the free app, or open **worksheets.ambiful.com/get**

Answer key

Queue · AW-12CS-QUE-2026-V1 · For parents — fold back or tear off before handing the sheet to your child.

Q	Answer	Working
1	**(d) First In, First Out (FIFO)**	—
2	**(a) [81, 63, 83]**	items leave from the front in the order they joined (FIFO)
3	**(a) 55**	dequeue() removes the item at the front: the earliest one still waiting
4	**(d) [94, 97]**	append() adds at the rear, insert(0, x) at the front; pop() removes from the rear, pop(0) from the front
5	**(d) stack**	—
6	**(c) 35**	the first 10 values leave; the next one is at the front
7	**(c) True**	pop(0) and pop() take the two ends; a palindrome matches all the way
8	**(b) 4**	9 − 8 + 3
9	**A linear data structure in which items join at the rear and leave from the front (FIFO); e.g. people in a line at a ticket counter.**	—
10	**28, 67, Underflow, 11, [21, 10]**	FIFO: each dequeue takes the earliest item still in the queue
11	**[99, 91, 95, 47, 87]**	append() adds at the rear, insert(0, x) at the front; pop() removes from the rear, pop(0) from the front
12	**True**	compare the front and rear letters until they meet
13	**13, 75**	the earliest item still waiting leaves first
14	**47, 65**	pop() takes the rear item, pop(0) the front item
15	**1**	the 1 value ahead of it must leave first
16	**stack**	plates piled one on another: stack
17	**q = []; enqueue: q.append(x); dequeue: if q: print(q.pop(0)) else print('Underflow'); peek: q[0] if q else 'Empty'; display: print(q). A while loop shows the menu.**	—
18	**(a) 71, 96; (b) [12, 26, 92]; (c) 12**	FIFO; dequeue() on an empty queue returns Underflow
19	**(a) [16, 23, 66], [23, 66], [23], [95, 23]; (b) [95, 23]; (c) append() and pop() (add and remove at the same end).**	append/pop work on the rear; insert(0, x)/pop(0) on the front
20	**(a) True; (b) pop(0) removes from the front and pop() from the rear, so the two ends are compared and removed until one or no letter is left; (c) True.**	a palindrome reads the same from both ends

Ambiful Worksheets · worksheets.ambiful.com · Not affiliated with CBSE / NCERT