Stack

Standard difficulty · 20 questions · 40 marks · Time: 42 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 stack works on the principle of:

(a) random access
(b) smallest first
(c) First In, First Out (FIFO)
(d) Last In, First Out (LIFO)

2. What is the last line this program prints?

```
def push(s, x):
    s.append(x)

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

st = []
push(st, 86)
push(st, 89)
push(st, 81)
print(pop(st))
push(st, 63)
print(pop(st))
push(st, 83)
print(st)
```

(a) [86, 89, 83]
(b) [89, 86, 83]
(c) [89, 83]
(d) [86, 83, 89]

3. What is the first line this program prints?

```
def push(s, x):
    s.append(x)

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

st = []
push(st, 55)
push(st, 58)
push(st, 75)
print(pop(st))
print(pop(st))
push(st, 91)
print(pop(st))
push(st, 19)
print(st)
```

(a) 75 **(b)** 19 **(c)** None **(d)** 55

4. What is the value of the postfix expression 7, 9, 8, +, * (tokens separated by commas)?

(a) 120 **(b)** 119 **(c)** 121 **(d)** 118

5. What is the postfix form of the infix expression (A + B) / C - D?

(a) AB+C/D- **(b)** ABC/+D- **(c)** ABCD+/- **(d)** -/+ABCD

6. Which of these bracket strings is balanced (every opening bracket closed by the matching one, in the right order)?

(a) {{{}}}{{[]}[]}()] **(b)** {{{}}}{{[]}[]}[](]

(c) {{{}}}{{[]}[]}[[)] **(d)** {{{}}}{{[]}[]}[()]

7. While the postfix expression 2, 8, 9, +, 5, *, * is evaluated with a stack, what is the largest number of values on the stack at one time?

(a) 2 **(b)** 3 **(c)** 4 **(d)** 5

8. How many items are on the stack at the end?

```
def push(s, x):
    s.append(x)

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

st = []
push(st, 62)
print(pop(st))
push(st, 28)
push(st, 67)
push(st, 11)
push(st, 21)
push(st, 10)
print(st)
```

(a) 5 **(b)** 6 **(c)** 1 **(d)** 4

Section B — Short answer 8 × 2 = 16 marks

9. What is a stack? Give one example from daily life.

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

```
def push(s, x):
    s.append(x)

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

st = []
push(st, 10)
push(st, 50)
push(st, 95)
push(st, 84)
print(pop(st))
print(pop(st))
push(st, 83)
print(st)
```

11. Evaluate the postfix expression 7, 3, 3, /, * (tokens separated by commas) using a stack.

12. Convert the infix expression A / (B + (C - D)) to postfix.

13. What does this print?

```
word = 'QUEUE'
st = []
for ch in word:
    st.append(ch)
rev = ''
while st:
    rev = rev + st.pop()
print(rev)
```

14. A stack is used to check the brackets in [[]([][])]{[()]}. Are they balanced? Answer Yes or No.

15. The postfix expression 8, 4, 2, +, −, 9, + is evaluated with a stack. Write the stack, bottom first, just after token 2 (4) is read, numbers separated by commas.

16. What is the top of the stack when this program ends? Write Empty if there is none.

```
def push(s, x):
    s.append(x)

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

st = []
print(pop(st))
push(st, 92)
print(pop(st))
push(st, 34)
push(st, 71)
push(st, 96)
print(pop(st))
push(st, 12)
push(st, 26)
print(st)
```

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

17. Write a menu-driven Python program to implement a stack using a list, with push, pop, peek and display.

18. Trace this program. (a) What does each print(pop(st)) print? Write them in order. (b) What is the final stack? (c) What is the top of the stack at the end?

```
def push(s, x):
    s.append(x)

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

st = []
push(st, 43)
push(st, 69)
push(st, 73)
push(st, 63)
push(st, 46)
push(st, 32)
push(st, 51)
print(pop(st))
print(st)
```

19. Convert A / B + C / D to postfix using a stack. Show the stack and the output after each symbol. (a) What is the postfix expression? (b) What is the largest number of symbols on the stack at one time?

20. (a) What does this program print? (b) Why does it pop b before a?

```
exp = ['3', '6', '+', '9', '9', '/', '/']
st = []
for t in exp:
    if t.isdigit():
        st.append(int(t))
    else:
        b = st.pop()
        a = st.pop()
        if t == '+':
            st.append(a + b)
        elif t == '-':
            st.append(a - b)
        elif t == '*':
            st.append(a * b)
        else:
            st.append(a // b)
print(st.pop())
```

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

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

Q	Answer	Working
1	**(d) Last In, First Out (LIFO)**	—
2	**(a) [86, 89, 83]**	the last item pushed is the first popped (LIFO)
3	**(a) 75**	pop() returns the item on top: the one pushed most recently
4	**(b) 119**	stack after each token: [7] [7, 9] [7, 9, 8] [7, 17] [119]
5	**(a) AB+C/D-**	operands go straight to the output; an operator waits on the stack until one of lower precedence (or a closing bracket) comes
6	**(d) {{{}}}{{[]}[]}[()]**	push each opening bracket; each closing bracket must match the one popped; the stack must end empty
7	**(b) 3**	stack after each token: [2] [2, 8] [2, 8, 9] [2, 17] [2, 17, 5] [2, 85] [170]
8	**(a) 5**	items pushed minus items popped (an Underflow pops nothing)
9	**A linear data structure in which items are added and removed only at one end, the top (LIFO); e.g. a pile of plates.**	—
10	**84, 95, [10, 50, 83]**	LIFO: each pop takes the most recent push
11	**7**	stack after each token: [7] [7, 3] [7, 3, 3] [7, 1] [7]
12	**ABCD-+/**	operands to the output; an operator first pops those of equal or higher precedence
13	**EUEUQ**	the characters come off the stack in reverse order
14	**Yes**	every closing bracket matches the top of the stack and the stack ends empty
15	**8, 4**	stacks: [8] [8, 4] [8, 4, 2] [8, 6] [2] [2, 9] [11]
16	**26**	the top is the last item pushed that was not popped
17	**st = []; push: st.append(x); pop: if st: print(st.pop()) else print("Underflow"); peek: print(st[len(st) − 1]) if st else "Empty"; display: for x in reversed(st): print(x). A while loop shows the menu and calls the chosen function.**	—
18	**(a) 51; (b) [43, 69, 73, 63, 46, 32]; (c) 32**	LIFO; pop() on an empty stack returns Underflow
19	**(a) AB/CD/+; (b) 2**	operands go to the output; (is pushed;) pops to the matching (; an operator pops those of equal or higher precedence first
20	**(a) 9; (b) the right operand was pushed last, so it comes off first; a − b and a // b need the operands in their original order.**	evaluates the postfix expression 3, 6, +, 9, 9, /, /

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