

السؤال 1: تتبع الكود

الكود يقوم بإضافة وإزالة العناصر من المكس. التتبع يكون كالتالي:

1. $\text{push}('A') : [A]$
2. $\text{push}('B') : [A, B]$
3. $\text{push}('C') : [A, B, C]$
4. $\text{pop}() : [A, B]$
5. $\text{pop}() : [A]$
6. $\text{push}('D') : [A, D]$
7. $\text{push}('E') : [A, D, E]$
8. $\text{push}('F') : [A, D, E, F]$
9. $\text{pop}() : [A, D, E]$
10. $\text{push}('G') : [A, D, E, G]$
11. $\text{pop}() : [A, D, E]$
12. $\text{pop}() : [A, D]$
13. $\text{pop}() : [A]$

السؤال 2: حساب حجم المكس وعدد العمليات

- عدد عمليات $\text{push} = 25$
- عدد عمليات $\text{pop} = 10$ (3) منها أعادت null لكون المكس فارغاً).
- إذا أرجعت null ، فهذا يعني أن المكس كان فارغاً أثناء 3 عمليات pop .

حجم المكس الحالي: الحجم = عدد عمليات الدفع - عدد عمليات الإزالة الصالحة | الحجم = عدد عمليات الدفع - عدد عمليات الإزالة الصالحة
 $= 25 - (10 - 3) = 25 - 7 = 18$

قيمة المتغير t (قمة المكس) = مؤشر آخر عنصر تمت إضافته.

السؤال 3: تقييم التعبيرات البعدية (Postfix Expressions)

a. $8 \ 2 \ + \ 3 \ * \ 16 \ 4 \ / \ -$

1. $2 \ + \ 8 \rightarrow 10$ جمع
2. $3 \ * \ 10 \rightarrow 30$ ضرب
3. $4 \div 16 \rightarrow 4$ قسمة
4. $4 \ - \ 30 \rightarrow 26$ طرح

الناتج النهائي 26 :

b. $12 \ 2 \ 5 \ 5 \ 1 \ / \ / \ * \ 8 \ 7 \ + \ -$

1. $1 \div 5 \rightarrow 5$ قسمة
2. $5 \div 5 \rightarrow 1$ قسمة
3. $1 \ * \ 2 \rightarrow 2$ ضرب
4. $2 \ * \ 12 \rightarrow 24$ ضرب
5. $7 \ + \ 8 \rightarrow 15$ جمع
6. $15 \ - \ 24 \rightarrow 9$ طرح

الناتج النهائي 9 :

c. $70 \ 14 \ 4 \ 5 \ 15 \ 3 \ / \ * \ - \ / \ 6 \ +$

1. $3 \div 15 \rightarrow 5$ قسمة
2. $5 \ * \ 4 \rightarrow 20$ ضرب
3. $20 \ - \ 14 \rightarrow -6$ طرح
4. $6667 \div -6 \rightarrow -11.6667$ (تقريباً) قسمة
5. $6 \ + \ 11.6667 \rightarrow -5.6667$ طرح

الناتج النهائي -5.6667 :

d. $3 \ 5 \ 6 \ * \ + \ 13 \ - \ 18 \ 2 \ / \ +$

1. $6 \ * \ 5 \rightarrow 30$ ضرب
 2. $30 \ + \ 3 \rightarrow 33$ جمع
 3. $13 \ - \ 33 \rightarrow 20$ طرح
 4. $2 \div 18 \rightarrow 9$ قسمة
 5. $9 \ + \ 20 \rightarrow 29$ جمع
- الناتج النهائي 29

السؤال 4: تحويل التعبيرات

a. تحويل infix إلى postfix

$$(A + B) * (C + D) - E \rightarrow A B + C D + * E - \cdot$$

b. تحويل infix إلى postfix

$$A - (B + C) * D + E / F \rightarrow A B C + D * - E F / + \cdot$$

برمجة الأولى باستخدام: Java

```
java
Copy code
Stack<Character> stack = new Stack<>();
    stack.push('A');
    stack.push('B');
char sumAB = stack.pop() + stack.pop(); // A + B
    stack.push(sumAB);
    stack.push('C');
    stack.push('D');
char sumCD = stack.pop() + stack.pop(); // C + D
    stack.push(sumCD);
char result = stack.pop() * stack.pop(); // (A + B) *
    (C + D)
    stack.push('E');
النتيجة النهائية result = stack.pop() - stack.pop(); //
```

السؤال 5: تعريف دالة `printListReverse`

هذه الدالة تستخدم مكدياً لعكس طباعة قائمة مرتبطة.

الحل بلغة Java

```
java
Copy code
public void printListReverse(Node head) {
    Stack<Node> stack = new Stack<>();
    Node current = head;

    // دفع كل عناصر القائمة إلى المكدي
    while (current != null) {
        stack.push(current);
        current = current.next;
    }

    // طباعة العناصر أثناء إزالتها من المكدي
    while (!stack.isEmpty()) {
        System.out.println(stack.pop().data);
    }
}
```

السؤال 6: عكس محتويات مكس باستخدام `pop()` و `push()`

الحل بلغة Java

```
java
Copy code
public static <E> void reverse(ArrayStack<E> stack) {
    Stack<E> tempStack = new Stack<>();

    // نقل كل العناصر إلى المكس المؤقت
    while (!stack.isEmpty()) {
        tempStack.push(stack.pop());
    }

    // إعادة العناصر إلى المكس الأصلي
    while (!tempStack.isEmpty()) {
        stack.push(tempStack.pop());
    }
}
```

السؤال 7: إزالة العنصر السفلي من المكس

الحل بلغة Java

```
java
Copy code
public static <E> E popBottom(LinkedStack<E> stack) {
    if (stack.isEmpty()) return null;

    E topElement = stack.pop();
    if (stack.isEmpty()) {
        return topElement; // إذا كان العنصر الوحيد
    } else {
        E bottom = popBottom(stack);
        stack.push(topElement);
        return bottom;
    }
}
```

السؤال 8: استرجاع العنصر الثاني من القمة

الحل بلغة Java

```
java
Copy code
public E topSecond() {
    if (size < 2) throw new
IllegalStateException("Stack has less than 2
elements.");

    إزالة القمة      E top = pop();           //
العنصر الثاني      E second = top();          //
إعادة القمة      push(top);                 //
return second;
}
```

السؤال 9: إزالة العنصر الثاني من القمة

الحل بلغة Java

```
java
Copy code
public E popSecond() {
    if (size < 2) throw new
IllegalStateException("Stack has less than 2
elements.");

    إزالة القمة      E top = pop();           //
إزالة العنصر الثاني E second = pop();          //
إعادة القمة      push(top);                 //
return second;
}
```

السؤال 10: استرجاع العنصر السفلي من المكس (LinkedStack)

الحل بلغة Java

```
java
Copy code
public E bottom() {
    if (isEmpty()) throw new
IllegalStateException("Stack is empty.");

    E topElement = pop();
    if (isEmpty()) {
        push(topElement);
        return topElement; //
    } else {
        E bottomElement = bottom();
        push(topElement);
        return bottomElement;
    }
}
```

العنصر السفلي الوحيد

السؤال 11: إزالة العنصر السفلي من المكديس (ArrayStack)

الحل بلغة Java

```
java
Copy code
    public E popBottom() {
        if (isEmpty()) throw new
IllegalStateException("Stack is empty.");

E[] newArray = (E[]) new Object[capacity];
        int originalSize = size;
        E bottomElement = data[0];

        // نقل العناصر باستثناء الأول
        for (int i = 1; i < originalSize; i++) {
            newArray[i - 1] = data[i];
        }

        data = newArray;
        size--;
        return bottomElement;
    }
```

السؤال 12: تحليل الكود المعطى

الكود:

```
java
Copy code
Stack<int> stack = new ArrayStack(10);
for (int i = 1; i <= 10; i++) {
    if (i % 3 != 0) {
        stack.push(i * 2);
    } else {
        stack.pop();
    }
}
```

التحليل:

1. العناصر المدفوعة إلى المكس:

$i = 1 \rightarrow \text{push}(2)$ ○
 $i = 2 \rightarrow \text{push}(4)$ ○
 $i = 3 \rightarrow \text{pop}()$ ○
 $i = 4 \rightarrow \text{push}(8)$ ○
 $i = 5 \rightarrow \text{push}(10)$ ○
 $i = 6 \rightarrow \text{pop}()$ ○
 $i = 7 \rightarrow \text{push}(14)$ ○
 $i = 8 \rightarrow \text{push}(16)$ ○
 $i = 9 \rightarrow \text{pop}()$ ○
 $i = 10 \rightarrow \text{push}(20)$ ○

المكس النهائي [2, 4, 8, 14, 20]:

2. قيم المتغيرات:

○ $\text{count} = 10$ عدد العمليات في الحلقة.
○ $\text{top} = 5$ إشارة إلى العنصر الأخير المضاف. (20):

3. نتيجة: $\text{top}()$ العنصر العلوي هو 20.

4. هل المكس ممتلئ؟

○ لا، لأن الحجم الحالي = 5 < السعة = 10.

5. إعادة المكس إلى الحالة الفارغة:

○ قم بتنفيذ $\text{pop}()$ حتى يصبح المكس فارغاً.