2. انتبهوا: لهذا السؤال صيغتان: بلغة **Java** في الصفحتين 4-5 ، وبلغة **C#** في الصفحتين 6-7.

<u>للّذين يحلُّون بلغة Java</u>

أ. أمامكم العمليّة mmm ، التي تتلقّى دَوْرًا – q فيه أعداد أكبر من 0 ، وعددًا صحيحًا – z أكبر من 0 .

```
public static boolean mmm (Queue<Integer> q, int z)
{
    q.insert (0);
    int num = q.head();
    int y = 0;
    while (q.head() > 0)
    {
        if (y < z)
        {
            if (q.head() == num)
            {
                y++;
            }
            else
            {
                num = q.head();
                y = 1;
            }
        }
        q.insert (q.remove());
    }
    q.remove();
    return y == z;
}
```

معطى الدَّوْر q من نمط صحيح:

رأس الدَّوْر							نهاية الدَّوْر
	1	3	1	1	1	2	

(1) تتبَّعوا، بواسطة جدول المتابعة الذي أمامكم العمليّة mmm(q , 4) ، واكتبوا ماذا تُعيد العمليّة.

الدَّوْر q	num	y	y < z	q.head() == num

(2) اشرحوا ماذا تنفّذ العمليّة mmm .

(3) ما هي تعقيدات زمن تشغيل العمليّة mmm ؟ علِّلوا إجابتكم.

(انتبهوا: تكملة السؤال في الصفحة التالية.)

ب. أمامكم العمليّة what ، التي تتلقّى دَوْرًا – q فيه أعداد أكبر من 0 ، وكِبَر الدَّوْر – n .

```
public static int what (Queue<Integer> q, int n)
{
    if (mmm (q, n))
        return n;
    return what (q, n − 1);
}
```

معطى الدَّوْر – q من نمط صحيح :

رأس الدَّوْر							نهاية الدَّوْر
	1	3	1	1	1	2	

(1) تتبَّعوا العمليّة what (q, 6) ، واكتبوا ماذا تُعيد العمليّة (لا حاجة لمتابعة العمليّة mmm) .
تحوي المتابعة، في كلّ استدعاء، قيمتَيْ q ، n والقيمة المُعادة .
أمامكم اقتراح لجدول متابعة (استعمال هذا الجدول ليس إلزاميًّا) .

الدَّوْر q المتلقَّى في العمليّة	قيمة n المتلقّاة في العمليّة	mmm (q, n) == true	القيمة المُعادة

(2) اشرحوا ماذا تنفّذ العمليّة what .

(3) ما هي تعقيدات زمن تشغيل العمليّة what ؟ علِّلوا إجابتكم .