

أ. أمامكم العملية Mmm ، التي تتلقى دُورًا - q فيه أعداد أكبر من 0 ، وعددًا صحيحًا - z أكبر من 0 .

```
public static bool Mmm (Queue<int> 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 الدُّور q من نمط صحيح :

رأس الدُّور	نهاية الدُّور
1	2
3	
1	
1	
1	

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

الدُّور q	num	y	y < z	q.Head() == num

(2) اشرحوا ماذا تنفذ العملية Mmm .

(3) ما هي تعقيدات زمن تشغيل العملية Mmm ؟ علِّلوا إجاباتكم .

ب. أمامكم العملية What ، التي تتلقَى دَوْرًا - q فيه أعداد أكبر من 0 ، وكَبِرَ الدَّوْر - n .

```
public static int What (Queue<int> 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	2
3	1
1	1
1	1

(1) تتبَّعوا العملية What (q, 6) ، واكتبوا ماذا تُعيد العملية ( لا حاجة لمتابعة العملية Mmm ) .

تحوي المتابعة، في كلِّ استدعاء، قيمَتَي q ، n والقيمة المُعادة .  
أمامكم اقتراح لجدول متابعة ( استعمال هذا الجدول ليس إلزامياً ) .

الدَّوْر q المتلقَى في العملية	قيمة n المتلقاة في العملية	Mmm (q, n) == true	القيمة المُعادة

(2) اشرحوا ماذا تنفَّذ العملية What .

(3) ما هي تعقيدات زمن تشغيل العملية What ؟ عللوا إجابتكم .