

1. This question involves the AppointmentBook class, which provides methods for students to schedule appointments with their teacher. Appointments can be scheduled during one of eight class periods during the school day, numbered 1 through 8. A requested appointment has a duration, which is the number of minutes the appointment will last. The 60 minutes within a period are numbered 0 through 59. In order for an appointment to be scheduled, the teacher must have a block of consecutive, available minutes that contains at least the requested number of minutes in a requested period. Scheduled appointments must start and end within the same period.

The AppointmentBook class contains two helper methods, isMinuteFree and reserveBlock. You will write two additional methods of the AppointmentBook class.

```
public class AppointmentBook
{
    /**
     * Returns true if minute in period is available for an appointment and returns
     * false otherwise
     * Preconditions: 1 <= period <= 8; 0 <= minute <= 59
     */
    private boolean isMinuteFree(int period, int minute)
    { /* implementation not shown */ }

    /**
     * Marks the block of minutes that starts at startMinute in period and
     * is duration minutes long as reserved for an appointment
     * Preconditions: 1 <= period <= 8; 0 <= startMinute <= 59;
     * 1 <= duration <= 60
     */
    private void reserveBlock(int period, int startMinute, int duration)
    { /* implementation not shown */ }

    /**
     * Searches for the first block of duration free minutes during period, as described in
     * part (a). Returns the first minute in the block if such a block is found or returns -1 if no
     * such block is found.
     * Preconditions: 1 <= period <= 8; 1 <= duration <= 60
     */
    public int findFreeBlock(int period, int duration)
    { /* to be implemented in part (a) */ }

    /**
     * Searches periods from startPeriod to endPeriod, inclusive, for a block
     * of duration free minutes, as described in part (b). If such a block is found,
     * calls reserveBlock to reserve the block of minutes and returns true; otherwise
     * returns false.
     * Preconditions: 1 <= startPeriod <= endPeriod <= 8; 1 <= duration <= 60
     */
    public boolean makeAppointment(int startPeriod, int endPeriod,
                                   int duration)
    { /* to be implemented in part (b) */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

- (a) Write the findFreeBlock method, which searches period for the first block of free minutes that is duration minutes long. If such a block is found, findFreeBlock returns the first minute in the block. Otherwise, findFreeBlock returns -1. The findFreeBlock method uses the helper method isMinuteFree, which returns true if a particular minute is available to be included in a new appointment and returns false if the minute is unavailable.

Consider the following list of unavailable and available minutes in period 2.

Minutes in Period 2	Available?
0–9 (10 minutes)	No
10–14 (5 minutes)	Yes
15–29 (15 minutes)	No
30–44 (15 minutes)	Yes
45–49 (5 minutes)	No
50–59 (10 minutes)	Yes

The method call findFreeBlock(2, 15) would return 30 to indicate that a 15-minute block starting with minute 30 is available. No steps should be taken as a result of the call to findFreeBlock to mark those 15 minutes as unavailable.

The method call findFreeBlock(2, 9) would also return 30. Whenever there are multiple blocks that satisfy the requirement, the earliest starting minute is returned.

The method call findFreeBlock(2, 20) would return -1, since no 20-minute block of available minutes exists in period 2.

Complete method findFreeBlock. You must use isMinuteFree appropriately in order to receive full credit.

```
/**
 * Searches for the first block of duration free minutes during period, as described in
 * part (a). Returns the first minute in the block if such a block is found or returns -1 if no
 * such block is found.
 * Preconditions: 1 <= period <= 8; 1 <= duration <= 60
 */
public int findFreeBlock(int period, int duration)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

- (b) Write the makeAppointment method, which searches the periods from startPeriod to endPeriod, inclusive, for the earliest block of duration available minutes in the lowest-numbered period. If such a block is found, the makeAppointment method calls the helper method reserveBlock to mark the minutes in the block as unavailable and returns true. If no such block is found, the makeAppointment method returns false.

Consider the following list of unavailable and available minutes in periods 2, 3, and 4 and three successive calls to makeAppointment.

Period	Minutes	Available?
2	0–24 (25 minutes)	No
2	25–29 (5 minutes)	Yes
2	30–59 (30 minutes)	No
3	0–14 (15 minutes)	Yes
3	15–40 (26 minutes)	No
3	41–59 (19 minutes)	Yes
4	0–4 (5 minutes)	No
4	5–29 (25 minutes)	Yes
4	30–43 (14 minutes)	No
4	44–59 (16 minutes)	Yes

The method call makeAppointment(2, 4, 22) returns true and results in the minutes 5 through 26, inclusive, in period 4 being marked as unavailable.

The method call makeAppointment(3, 4, 3) returns true and results in the minutes 0 through 2, inclusive, in period 3 being marked as unavailable.

The method call makeAppointment(2, 4, 30) returns false, since there is no block of 30 available minutes in periods 2, 3, or 4.

The following shows the updated list of unavailable and available minutes in periods 2, 3, and 4 after the three example method calls are complete.

Period	Minutes	Available?
2	0–24 (25 minutes)	No
2	25–29 (5 minutes)	Yes
2	30–59 (30 minutes)	No
3	0–2 (3 minutes)	No
3	3–14 (12 minutes)	Yes
3	15–40 (26 minutes)	No
3	41–59 (19 minutes)	Yes
4	0–26 (27 minutes)	No
4	27–29 (3 minutes)	Yes
4	30–43 (14 minutes)	No
4	44–59 (16 minutes)	Yes

Complete method makeAppointment. Assume that findFreeBlock works as intended, regardless of what you wrote in part (a). You must use findFreeBlock and reserveBlock appropriately in order to receive full credit.

```
/**
 * Searches periods from startPeriod to endPeriod, inclusive, for a block
 * of duration free minutes, as described in part (b). If such a block is found,
 * calls reserveBlock to reserve the block of minutes and returns true; otherwise
 * returns false.
 * Preconditions: 1 <= startPeriod <= endPeriod <= 8; 1 <= duration <= 60
 */
public boolean makeAppointment(int startPeriod, int endPeriod,
                               int duration)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

Class information for this question

```
public class AppointmentBook
private boolean isMinuteFree(int period, int minute)
private void reserveBlock(int period, int startMinute,
                           int duration)
public int findFreeBlock(int period, int duration)
public boolean makeAppointment(int startPeriod, int endPeriod,
                               int duration)
```