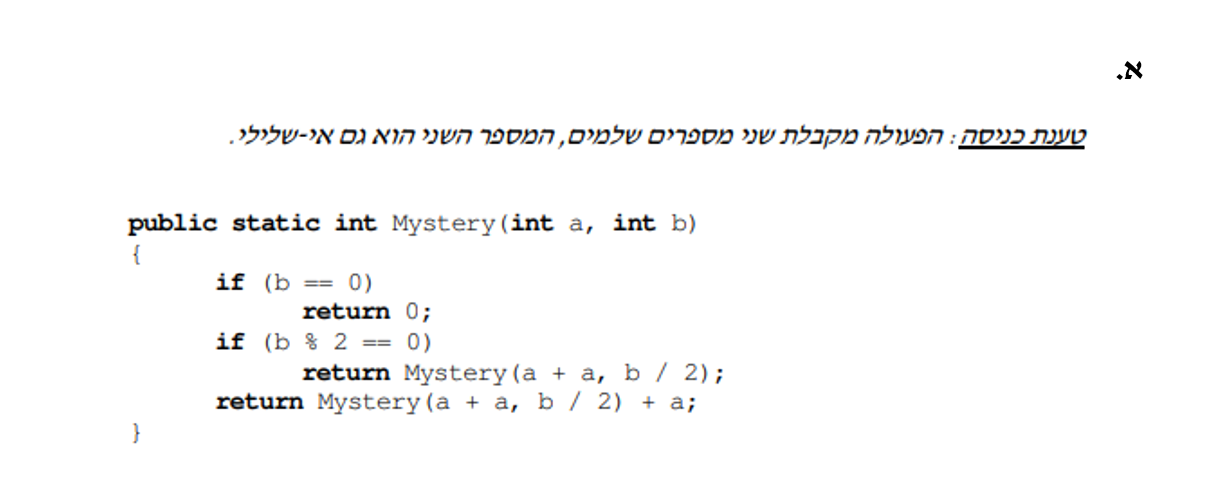
שאלה 4

א.- 

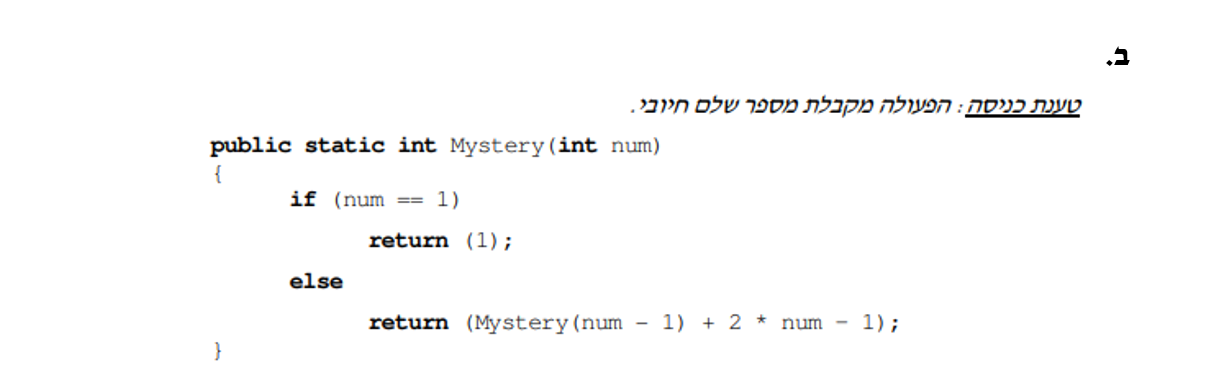
טענת יציאה: אם b שווה 0.

המטרה היא לכפול את b ב a.

אם b זוגי אז תחלק אותו בשתיים אבל תכפיל את a ב 2

אם אי זוגי אז תעשה אותו דבר ותתחיל לחבר את a . בסופו של דבר זה יגיע למספר אי זוגי ויחבר את a כמו מספר הפעמים של b.

b צריך להיות אי שלילי משום שאם שלילי אז זה לא משפיע על הסימן כל התפקיד של b זה לשמש כ count.

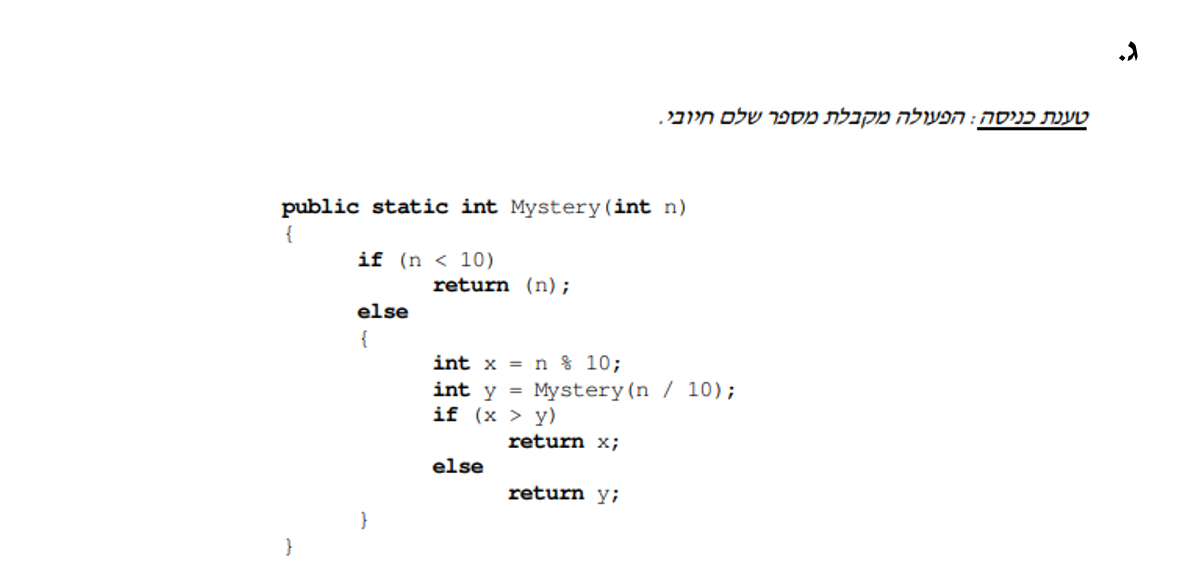
ב.-

טענת יציאה: אם המספר שווה ל -1

המטרה היא לעלות את המספר בריבוע.

ברגע שעברנו מספר פעמים בגודל המספר נחזיר 1. לאחר מכן נכפיל את כל המספרים עד המספר כולל ב2 ונחסר את המספר עצמו פחות 1. ככה שיצא

ג.-

תנאי יציאה – אם n קטן מ10

מטרת הקוד זה להחזיר את הספרה הכי גדולה במספר .

בעצם הקוד מגיע לספרה האחרונה ואז אם הספרה שלפני גדולה יותר הוא מחזיר אותה אם לא אז את זאתי. ככה עד שהוא מגיע לספרה הראשונה. כך המקסימלי נבחר.

**שאלה 5**

ב.- הפעולה בודקת האם המספר X נמצא במספר NUM.

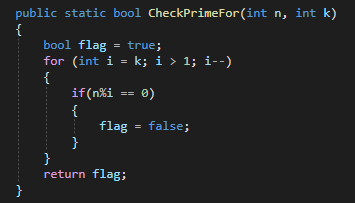
ד.- הפעולה מוסיפה ל K 1 אם המספר הקודם למספר במערך מוכל בתוכו.

**שאלה 6**

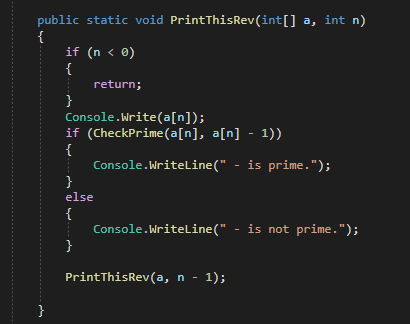
ב.- המתודה בודקת האם N הוא כפולה של K. היה ניתן להחליף קריאה זו בשורה של

ד.- היא בודקת אם N לא מתחלק בכל המספרים בין 2 ל K כולל(שניהם).

ניתן היה לכתוב:



ו.- ניתן לשנות ככה:



תרגיל 12 טורים

|  |  |
| --- | --- |
| תיאור | מתודה |
| המתודה מקבלת תאריך. היא עוברת על כל טורי המוצרים שבחנות וקוראת ל – SortQueue עבור כל אחד מהם. | public void ArrangeProducts(DateTime date) |
| המתודה מקבלת טור של מוצרים וגם תאריך.  המתודה מסדרת את טור המוצרים לפי תאריך פג התוקף שלהם. כאשר הכי קרוב להתקלקל בקדמת הטור. | private MyQueue<Product> SortQueue(MyQueue<Product> products, DateTime date) |
| מקבלת תאריך וטור של מוצרים.  מוציאה את כל המוצרים אשר תאריך פג התוקף שלהם הם לפני התאריך שהועבר כלומר פג תוקף. | private void RemoveOutdated(DateTime date,MyQueue<Product> products) |
| מקבלת טור ומחזירה את אורכו | public int GetSize(MyQueue<Product> queue) |

הדרך הכדאית ללקוח זה אם הוא היה מסדר הפוך כלומר לפי זה שהכי פחות קרוב לפג תוקף. הייתי פשוט מסדר הפוך.