**תרגול 1**

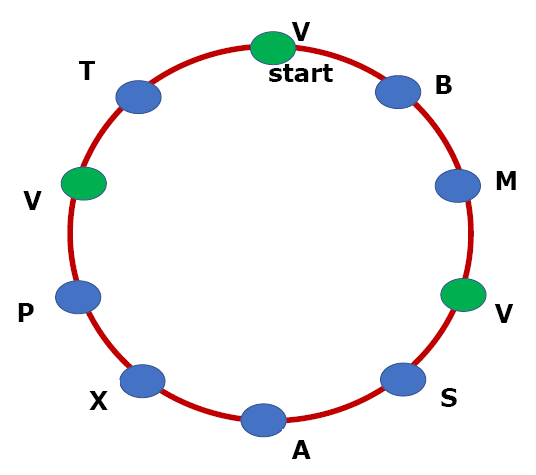
**בעיה 1: בעיית חנייה - חישוב מספר מכוניות בחנייה מעגלית.**

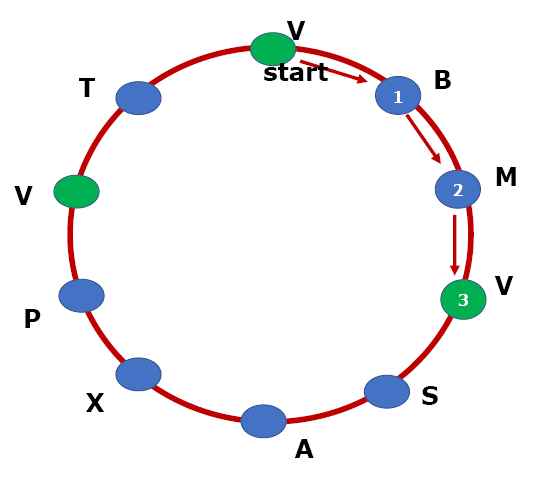
**תיאור הבעיה:** על החוקר לספור כמה מכוניות יש בחניה מעגלית.

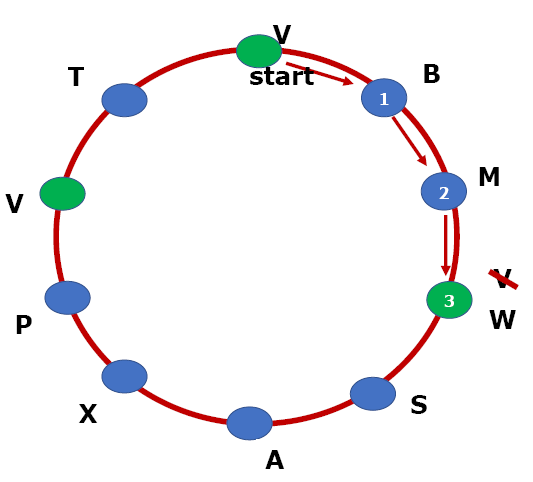
אורך המעגל אינו ידוע לחוקר.   
החנייה גדולה והחוקר רואה רק את המכונית שנמצאת לידו ואת המכונית הבאה.  
לחוקר ניתן לסמן את המכונית בסימן כלשהו, אך הסימן יכול כבר להופיע על מספר מכוניות (זאת אומרת שסימן לא ייחודי) .   
גם ניתן לחוקר למחוק את הסימן הקודם ולכתוב סימן חדש.

**אלגוריתם לפתרון הבעיה:**

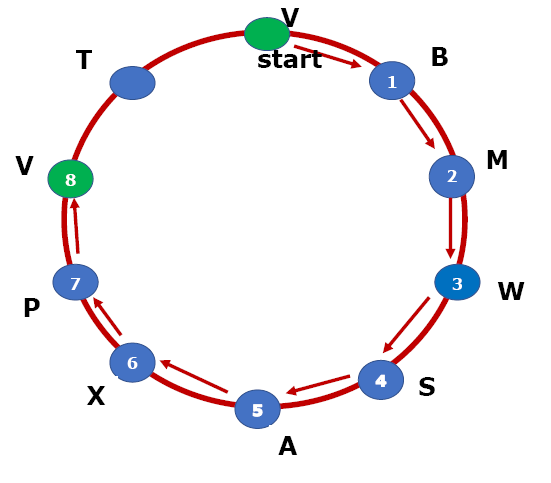
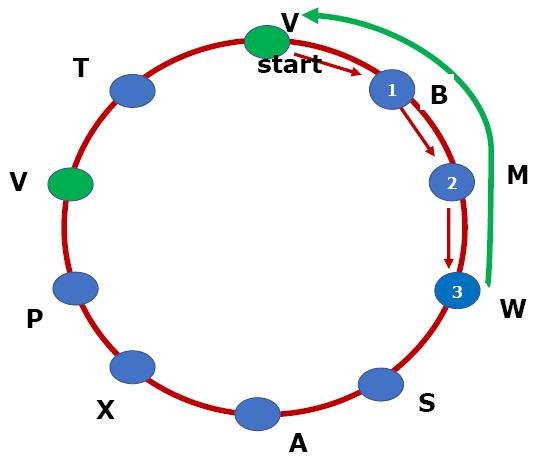
1. מכונית כלשהי מגדירים כראשונה ומסמנים אותה בסימן V (start).
2. עוברים על המעגל עד שמוצאים את המכונית עם הסימן V וסופרים את מספר מכוניות שעברנו - counter.
3. כשמגיעים למכונית עם סימן V, מחליפים את הסימן לסימן W.
4. חוזרים אחורה לנקודת ההתחלה (start). מספר צעדים אחורה שווה למספר צעדים קדימה שספרנו בשלב 2 (counter).
   1. אם רואים את W - מעגל סגור ובעיה נפתרה.
   2. אם רואים את V - הולכים קדימה לאותו מספר צעדים שחזרנו (counter) וממשיכים ללכת קדימה כוונה חוזרים לשלב 2.
5. מחזיריםcounter

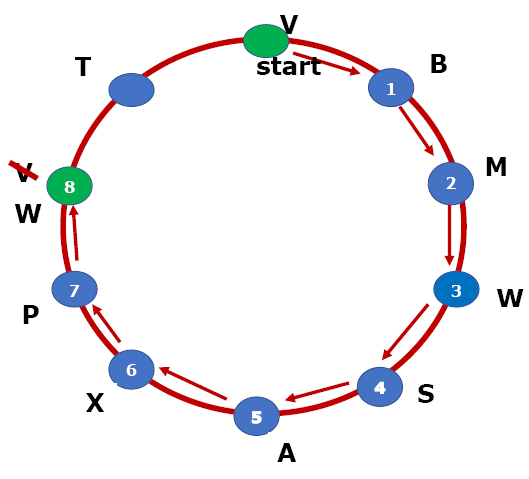
****

****

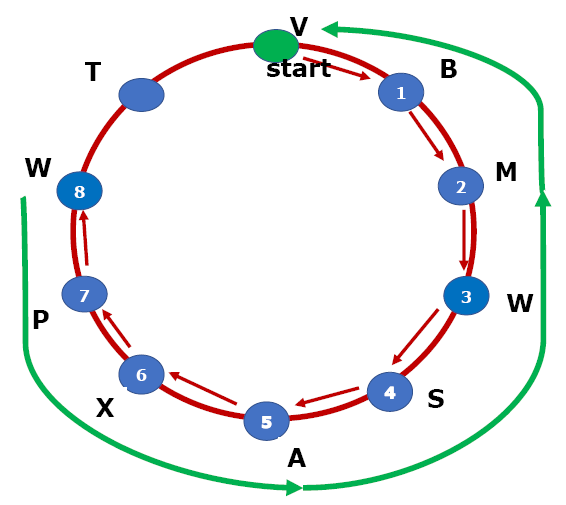
****

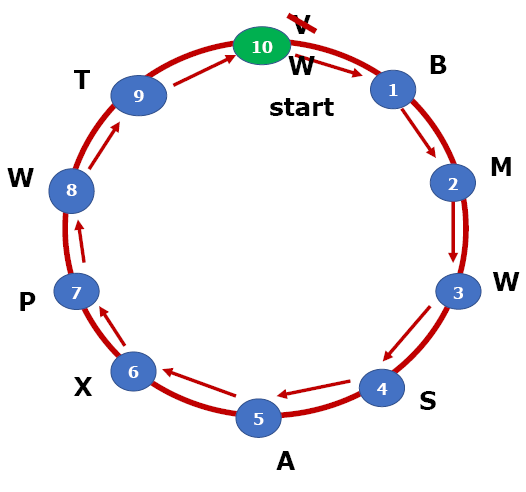
חזרה ל- V) **start** – המשך התהליך:**(**

****

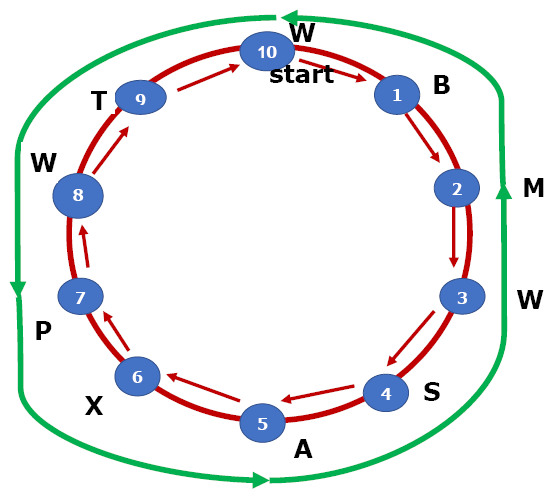
****

חזרה ל- V) **start** – המשך התהליך:**(**

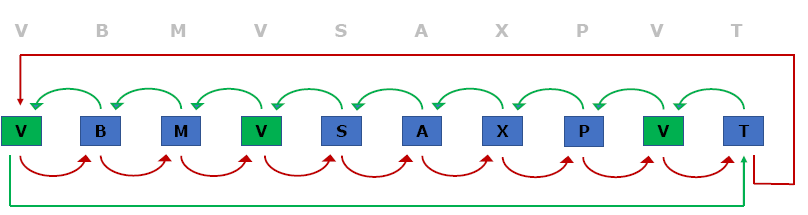
****

****

חזרה ל- W) **start** – סוף התהליך:**(**

****

מבנה נתונים לפתרון הבעיה: **רשימה מקושרת דו-כיוונית** **מעגלית**

****

**Variables**

DoubleCycleLinkedList **cars**

Node **currentNode**

int **count**

boolean **flag**

char **oldSign, newSign**

**Pseudocode**

**Complexity: O(n^2)**

**Initialization**

DoubleCycleLinkedList **cars** // Parking Double Cycle Linked

Node currentNode = **cars**.getHead().getNext() // initially - next to the head

boolean **flag** = true // continue LOOP

int **count** = 1 // the total number of cars

char **oldSign** = ‘V’, **newSign** = ‘W’ // the old & new signs

int **steps** = 0 // the number of steps back

main loop ( **while**(flag) ) // loop while begin

**if** currentNode.getData() ≠ **oldSign** // usual node : **no sign ‘V’**

currentNode = currentNode.getNext() // go to the next node

count++ // increase the counter

**else** // node with sign ‘**V**’

currentNode.setData(**newSign**) // set new sign ‘**W**’

steps = count

loop ( **while**(steps > 0) ) // loop for **steps** back

currentNode = currentNode.getPrev()// get previous node

steps-- // decrease the step

loop **while** end

**if** currentNode.getData() == **newSign**// back to new sign

flag = **false**

**else**

count = 1

currentNode = **cars**.getHead().getNext()

main loop **while** end