

# שאלה 1:

לפניך שתי פעולות חיזוניות.

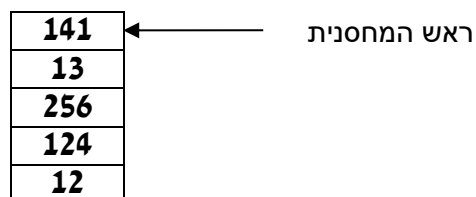
// הפעולה מקבלת מחסנית לא ריקה המכילה מספרים שלמים וחיוביים  
 // הפעולה מחזירה...

```
public static int Mystery1(Stack<int> s)
{
    int x = s.Pop() ;
    int y = x ;
    if(!s.IsEmpty())
    {
        int z = Mystery1(s) ;
        if(y<z)
            y=z ;
    }
    s.Push(x) ;
    return y ;
}
```

// הפעולה מקבלת מספר שלם וחיובי  
 // הפעולה מחזירה ...

```
public static int Mystery2(int num)
{
    if (num<10)
    {
        if (num%2==0)
            return num+1 ;
        else
            return num ;
    }
    else
    {
        if (num%2==0)
            return Mystery2(num/10)*10+(num%10+1) ;
        else
            return Mystery2(num/10)*10+(num%10) ;
    }
}
```

נתונה המחסנית st מטיפוס Stack<int>



- א. מה יחזיר הזימון `mystery1(st)` ? רשום את המעקב.
- ב. מה מבצעת הפעולה `mystery1(stack)` בעבור מחסנית `stack` לא ריקה מטיפוס `Stack<int>` ? רשום את המעקב.
- ג. מה מבצעת הפעולה `mystery2(k)` בעבור מספר `k` חיובי מטיפוס שלם.
- ד. מה מבצעת הפעולה `mystery2(mystery1(stack))` בעבור מחסנית `stack` לא ריקה מטיפוס `Stack<int>` המכילה מספרים שלמים וחיוביים.

## שאלה 2:

- א. כתוב פעולה `public static Queue<int> Arrange (Queue<int> q)` הפעולה מקבלת תור `q`, ומפזרת אותו לתורים חדשים `y` ו `z` - לפי האלגוריתם הבא:  
האיבר הראשון מוכנס לתור `y` ומכאן ואילך - אם האיבר בראש תור `q` גדול מהאיבר בסוף תור `y` הוא מוכנס לתור `y`, אחרת הוא מוכנס לתור `z`.  
לבסוף מעתיקים תור `y` ל `q` - ומחזירים את התור `z`.  
ב. נתונה הפעולה הבאה:
- ```
public static Queue<int> Merge (Queue<int> q1, Queue<int> q2);
```
- הפעולה `Merge` מקבלת `q1` ו `q2` - וממזגת אותם לתור `x`, לפי האלגוריתם הבא: משווים את שני האיברים בראשי התורים `q1` ו `q2`, -ומכניסים לתור `x` את האיבר הקטן מביניהם. הפעולה נמשכת עד אשר אחד התורים מתרוקן.  
הפעולה בסוף מחזירה את התור `x`.  
נתונה הפעולה `Mystery`:

```
public static Queue<int> Mystery(Queue<int> q)
{
    bool sod = false;
    while (!sod)
    {
        Queue<int> temp = Arrange(q);
        if (temp.IsEmpty())
            sod = true;
        else
            q = Merge(q, temp);
    }
    return q;
}
```

- יש לבצע מעקב אחר הפעולה. `mystery` עבור התור: 15,10,2,21,3
- במעקב יש לציין כיצד יראה התורים `temp` ו `q` - בכל אחד מהכניסות ללולאה
- ג. הגדר טענת יציאה עבור הפעולה `Mystery`
- ד. מה יהיה סדר גודל זמן ריצה של פעולת `mystery`

## שאלה 3:

- סטודנטים הממתינים לקבלת מלגה מאוחסנים בתור, הכולל את המידע הבא: מספר סטודנט, תואר (BA או MA), ממוצע ציונים והאם אזרח?
- זכאים למלגה אוטומטית הם אזרחי המדינה שממוצע ציונם מעל 90 ב BA - או מעל 85 ב MA -
- א. הגדר מחלקה מתאימה. אין צורך לממש פעולות. `set`, `get`
- ב. כתוב פעולה המקבלת את התור ומחזירה כמה זכאים אוטומטית למלגה ( אין להרוס את התור המקורי)
- ג. כתוב פעולה המקבלת סכום מלגה שאותו ניתן לחלק לסטודנטים באופן הבא:
- אם נמצאו 5 זכאים או יותר אוטומטית הסכום יחולק שווה בשווה ביניהם
  - אם נמצאו פחות, יחולק הסכום באופן הבא 50%: יחולק בין הזכאים אוטומטית ו 50% נוספים
- בין כל. אלה שאינם אזרחי המדינה, אך ממוצע ציונם מעל 95
- כל אלה שיקבלו מלגה יודפס הודעה עם שם הסטודנט והסכום שקיבל ויצא מהתור. חובה להשתמש בסעיף א'.
- ד. כתוב סדר גודל זמן ריצה של הפעולה שכתבת בסעיף ג'

## שאלה 4

במשרד הסחבת משתרך תמיד תור ארוך של ממתנים המורכב משני מסלולים: אדום וירוק .

במסלול הירוק ממתנים אנשים שצריכים להגיש לפקיד טפסים בלבד . במסלול האדום ניצבים אלו שלפניהם שהות ארוכה אצל הפקיד: תלונות, דיונים ועוד .

אנשים המגיעים למשרד נכנסים למסלול המתאים. בכל פעם שהפקיד מתפנה הוא מכריז: "הבא בתור!". הכרזה זו גוררת אחריה יציאה של אדם מהתור: אם המסלול הירוק אינו ריק, יצא ממנו האדם העומד בראשו וניגש לפקיד לקבל שרות. אחרת, ניגש לפקיד אדם מראש המסלול האדום. עבור כל ממתין נשמרים הנתונים הבאים: שם, ת.ז. וטלפון .

לפניך ממשק המחלקה מאגר-ממתנים WaitingList :

|                             |                                                                                                                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| אתחל-מאגר-ממתנים            | פעולה המאתחלת מאגר-ממתנים להיות מאגר ריק                                                                                                                                              |
| הכנס-למאגר-ממתנים( $x, k$ ) | פעולה המכניסה את הממתין $x$ לסוף התור המתאים (מהיר או איטי) על פי ערכו של $k$<br>הנחה: הנתונים תקינים                                                                                 |
| הוצא-ממאגר-ממתנים()         | פעולה המוציאה ומחזירה את הממתין שהוצא מהמאגר לפי מדיניות הקבלה לפקיד: אם התור המהיר אינו ריק, מוחזר הממתין הראשון בתור זה, אחרת מוחזר הממתין הראשון בתור האיטי.<br>הנחה: המאגר לא ריק |
| מאגר-ממתנים-ריק()?          | פעולה המחזירה 'אמת' אם המאגר-ממתנים, על שני תוריו ריק, ו-'שקר' אחרת.                                                                                                                  |
| תיאור-מאגר-הממתנים          | פעולה המחזירה מחרוזת המתארת את מאגר הממתנים                                                                                                                                           |

1. כתוב את המחלקה ממתין - Waiting (5 נקודות)
2. כתוב את המחלקה מאגר-ממתנים (WaitingList), וממש בתוכה את פעולות הממשק. (10 נקודות)
3. כתוב תכנית לבדיקת הפעולות שכתבת. על התכנית ליצור לפחות 5 ממתנים משני הסוגים ולאחר מכן מדמה את פעולת הפקיד ומראה מהו סדר הוצאת הממתנים מהתור בקריאה: "הבא בתור". הצג את המאגר אחרי כל שלב. (10 נקודות)

#### שאלה 4:

מעבורת של נוסעים נוסעת בין איים שונים. המחלקה שמייצגת אותה:

```
public class Ferry
{
    private int cap; // תכולה מקסימלית של כמות נוסעים
    private Queue<Integer> pass; // אוסף מספרי הזהות של הנוסעים שכרגע על המעבורת
```

מחלקת אי מוגדרת כך:

```
public class Island {
    private String name; // שם אי
    private Queue<Integer> waiting; // תור ממתינים למעבורת
```

א – כתבו במחלקת מעבורת פעולה שמקבלת אי אליו היא הגיעה. על הפעולה להוריד קודם את כל הנוסעים ולהעלות את הממתינים על האי מבלי לעבור את המכסה המקסימלית של הנוסעים, כמו כן אין להעלות נוסעים שזה עתה ירדו וחובה לשמור אותם בתור הממתינים של אי זה.

ב – כתבו פעולה במחלקת מעבורת שמקבלת מערך של כל האיים בקו הנסיעה של המעבורת ומספר זהות של נוסע. על הפעולה לעבור בין האיים ולבצע החלפת נוסעים, עד שמגיעים לאי בו הנוסע הרצוי נמצא. על הפעולה להדפיס את שם האי ולהחזיר אמת, אם הנוסע הצליח לעלות על המעבורת, אחרת תחזיר שקר.

ניתן להניח שמספר הזהות קיים באחד האיים ולא באחרון (שם מתבצעת הורדה בלבד).

ג – חשבו את זמן סיבוכיות הריצה של פעולה ב'. חובה לנמק.