

תכנות מונחה עצמים**עבודת הגשה 2**

מועד הגשה: 7.4.2019 בשעה 23:50

הוראות הגשה:

1. **אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.**
2. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
3. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
4. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
5. למחלקות המפורטות בעבודה ניתן להוסיף מתודות נוספות אך הם חייבות להיות private.
6. יש למקם כל מחלקה שיהיה עליכם ליצור, בשני קבצים נפרדים H ו-CPP, כאשר לכל תרגיל יהיה driver משלו. יש להכניס את החלק התיאורטי בקובץ וורד נפרד. יש להכניס את כל הקבצים של תרגיל 1 לתיקיה בשם Ex1, את כל הקבצים של תרגיל השני לתיקיה בשם Ex2, ואז לכווץ יחד עם קובץ וורד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.
7. **שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית לתרגיל, סבטלנה רוסיין, במייל: sceassign2016@gmail.com.**

חלק א' – תאורטי (מענה בקובץ טקסט – וורד): 7 נקודות

1. ציינו מה הם היתרונות של שימוש באופרטורים "new" ו-"delete" במקום "malloc" ו-"free"?
2. באיזה מקרים הקומפיילר יוצר אובייקטים זמניים. תן לפחות 2 דוגמאות.
3. האם ניתן להעמיס (overloaded) שתי מתודות זהות כאשר אחת מהן מוגדרת עם const והשנייה ללא?
4. האם משתנה סטטי של המחלקה יכול להיות מסוג const?
5. מדוע כשאנו שולחים אובייקט לפונקציה/מתודה כפרמטר עדיף לשלוח אותו בתור by const reference מאשר להעבירו בתור by value?
6. האם הטענה הבאה היא טענה נכונה: מתודה שמוגדרת כ-const לא יכולה להפעיל על אותו this מתודה שהיא לא מוגדרת כ-const. תן נימוק קצר.
7. האם הטענה הבאה היא טענה נכונה: לא ניתן להפעיל מתודה סטטית ללא שימוש באובייקט. תן נימוק קצר.

חלק ב' – מעשי (ההגשה היא של קבצי ה-CPP ו-H בלבד):

תרגיל מס' 1 **43 נקודות:** (בתרגיל הזה - יש להגיש סה"כ 7 קבצים – 2 קבצים לכל מחלקה (cpp ו-h) – cpp לדרייבר

תיאור העבודה:

בעבודה הזאת אתם מתבקשים לממש משחק "פרים" ו"פרות"
http://en.wikipedia.org/wiki/Bulls_and_cows

המשחק מתנהל בין המחשב לבין השחקן (משתמש). המחשב "יוזם" (בוחר באופן אקראי) מספר 4-ספרתי המורכב מספרות שונות ללא אפס. מטרת השחקן לנחש המספר זה עם כמות ניסויים הכי הקטנה שאפשר. בכל ניסוי המחשב מודיע על הצלחת ניחוש באמצעות כמות "פרות" ו"פרים". משמעות "פרה" (Cow) היא הספרה הנכונה הנמצאת במספר במיקום לא נכון. משמעות "פר" (Bull) היא הספרה הנכונה הנמצאת במספר במיקום נכון.

האילוצים לקלט של המשתמש:

- ✓ הקלט הנכון חייב לכלול רק ספרות.
- ✓ מספר שמקליד המשתמש חייב להיות 4-ספרתי.
- ✓ המשתמש לא יכול להקליד ספרה אפס או מספר בעל ספרות זהות. במידה והקלט היה לא תקין יש לדרוש מהמשתמש הזנה חוזרת.

תהינה מחלקות הבאות:

1. מחלקת ספרה Digit

המכילה את השדה ערך: value מטיפוס int

יש להגדיר במחלקה הזו מתודות:

- בנאי ברירת המחדל המאפס את הערך (כלומר מכניס 0 לערך)
- מתודה בוליאנית להשוות ספרה של האובייקט עם ספרה אחרת (אותה היא מקבלת בתור הפרמטר שלה).
- מתודה המחזירה את הערך השמור בתוך האובייקט.
- מתודה המקבלת מספר המייצג ספרה בודדת ומכניס אותו לשדה ערך של האובייקט.

2. מחלקת מספר Number

המכילה: מערך של 4 אובייקטים מסוג Digit

יש להגדיר במחלקה הזו מתודות:

- בנאי ברירת המחדל המאפס את כל הספרות של המספר (כלומר מכניס 0 לכל הספרות).
- מתודה המאפשרת למשתמש להכניס את המספר לתוך האובייקט.
- מתודה המגרילה את המספר כאשר נדרש להגריל כל ספרה בנפרד.
- מתודה המקבלת מיקום של הספרה בתוך המספר ומחזירה את הספרה הנמצאת במיקום זה, במידה המיקום לא תקין היא תחזיר -1.
- מתודה המדפיסה את המספר.

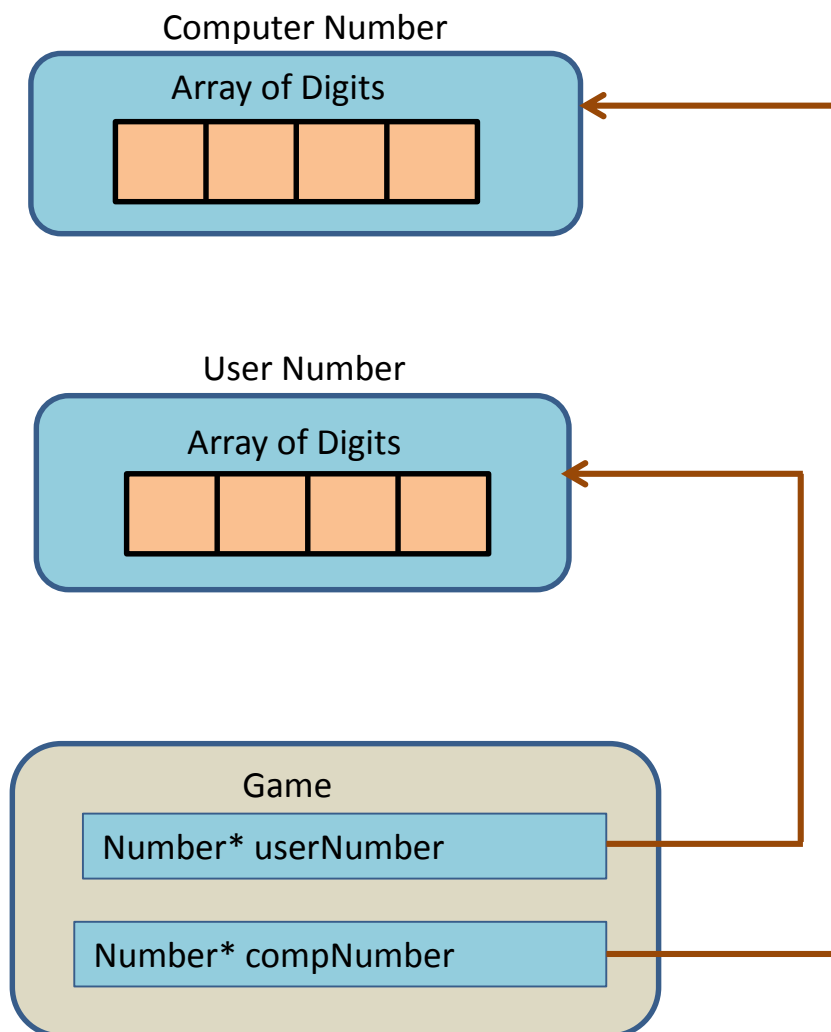
3. מחלקת משחק Game

המכילה את שני מצבעים מסוג Number:

אחד – למספר של מחשב ושני – למספר המשתמש.

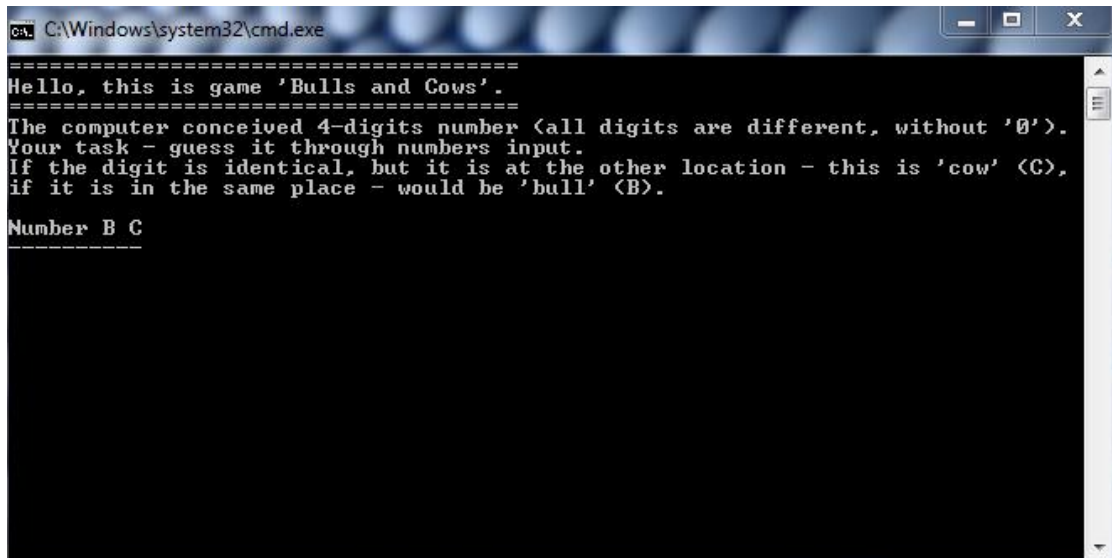
יש להגדיר במחלקה הזו מתודות ציבוריות :

- בנאי – לאיתחול המספרים .
- הורס
- בנאי מעתיק
- מתודה `play()` להפעלת המשחק .
- יש להגדיר במחלקה הזו מתודות פרטיות :
- מתודה המחזירה כמות "פרות".
- מתודה המחזירה כמות "פרים".



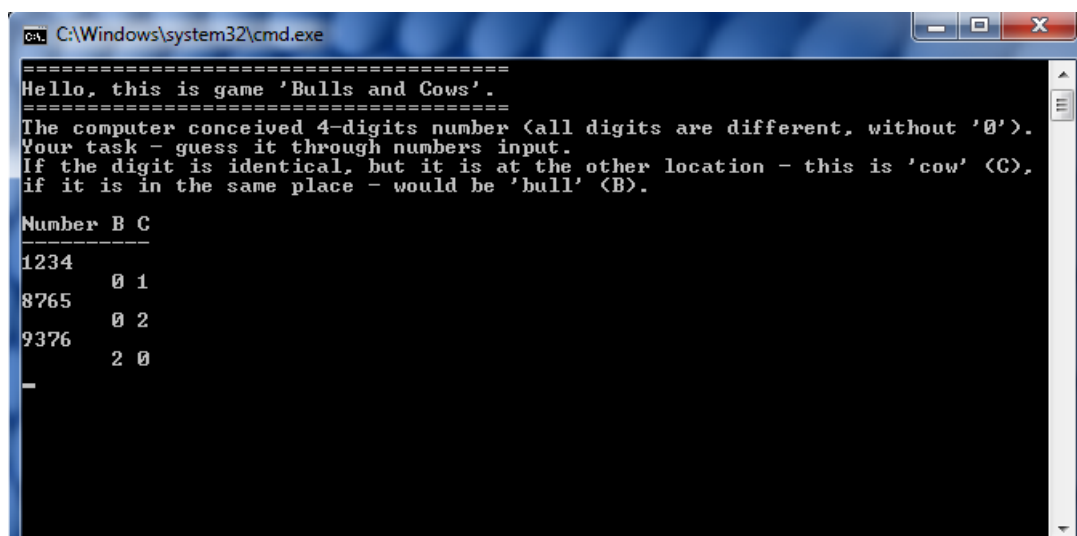
בתחילת משחק המחשב "יוזם" מספר 4-ספרתי. מטרת המשחק לנחש המספר ההוא עם כמות ניסויים הקטנה ביותר.

איור מס' 1:



אחרי כל ניסוי המחשב מודיע על הצלחת הניחוש באמצעות כמות "פרות" (C) ו"פריים" (B). משמעות "פרה" היא הספרה הנכונה הנמצאת במספר במיקום לא נכון. משמעות "פר" היא הספרה הנכונה הנמצאת במספר במיקום נכון. להלן התוצאות של שלוש ניסויים של המשתמש:

איור מס' 2:



Number	B	C
1234	0	1
8765	0	2
9376	2	0

בכל קלט של המשתמש יש לבדוק את נכונותו. הודעת שגיאה מודפסת גם במקרה שהמשתמש מנסה להכניס מספר יותר גדול מ-9999.

בסוף המשחק המחשב מודיע על 4 "פרים" – סימן שהמשתמש ניחש מספר נכון: איור מס' 3:

```

C:\Windows\system32\cmd.exe
=====
The computer conceived 4-digits number (all digits are different, without '0').
Your task - guess it through numbers input.
If the digit is identical, but it is at the other location - this is 'cow' (C),
if it is in the same place - would be 'bull' (B).

Number B C
-----
1234
      0 1
8765
      0 2
9376
      2 0
9342
      2 0
9358
      1 1
9846
      4 0
*****
YOU WON !!!
*****
New game? (Y/N)
  
```

תרגיל מס' 2 50 נקודות: (בתרגיל הזה - יש להגיש סה"כ 5 קבצים – 2 קבצים לכל מחלקה
cpp - i) cpp לדרייבר

תיאור העבודה:

בעבודה זו אתם תתכנתו גירסה של משחק הנייר ועיפרון המפורסם – צוללות (באנגלית Battleships) את התיאור של המשחק ניתן לקרוא:
[http://en.wikipedia.org/wiki/Battleship_\(game\)](http://en.wikipedia.org/wiki/Battleship_(game))

במשחק משתתף רק שחקן אחד. המחשב רק מציב את הצוללות והשחקן מנחש. בחלק הראשון של העבודה כתבו מחלקה המייצגת צוללת אחת. בחלק השני כתבו מחלקה המייצגת לוח משחק. בחלק השלישי כתבו את התוכנית הראשית.

חלק ראשון :

יש להגדיר מחלקה בשם **Battleship** המתארת צוללת (ספינת קרב). משתני המחלקה הם:

- אורך הצוללת.
- משתנה סטטי שיעזור בקביעת האורך של הצוללת (המשתנה מייצג את מספר הצוללות שכבר מוקמו על הלוח).

יש להגדיר את המתודות הבאות למחלקה:

- בנאי ברירת מחדל – הבנאי יבנה צוללת באורך שיקבע (בעזרת משתנה סטטי אשר יאותחל ב-0).

אורך הספירות יקבע לפי "מספרן בתור" :

- צוללת 1 (הצוללת שתבנה ראשונה) – אורך 5.
- צוללת 2 (הצוללת שתבנה שנייה) – אורך 4.
- צוללת 3 – אורך 3.
- צוללת 4 – אורך 2.
- צוללת 5 – אורך 3.

אורך צוללת לא יעלה על 5 ולא יפחת מ-2.

- לכל משתנה שזה רלוונטי עבורו – פונקציות גישה . (set/get) יש להימנע משיטות גישה לא נחוצות.

תיאור העבודה – חלק שני:

חלק שני :

יש להגדיר מחלקה בשם **Game** המתארת לוח משחק (שימו לב שמדובר על משחק אחד!)

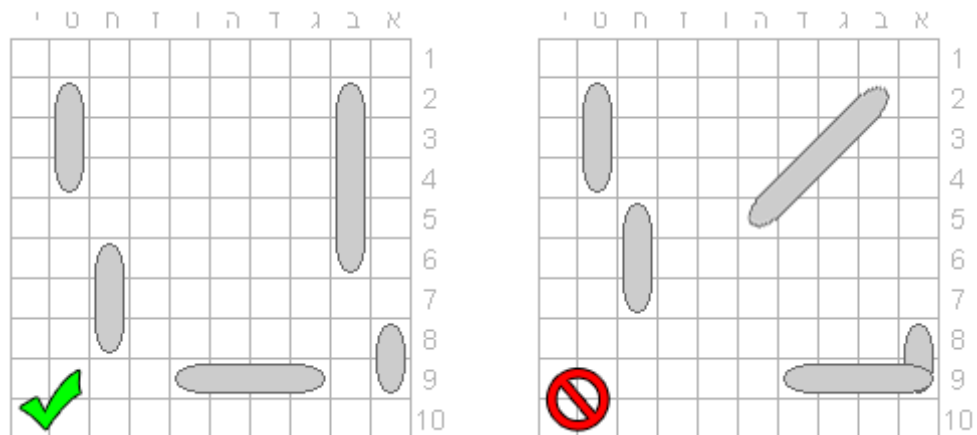
משתני המחלקה הם:

- ✓ לוח משחק בגודל 10*10 מסוג char
- ✓ מערך דינמי של צוללות (מצביע ל Battleship **ומספר הצוללות**).
- ✓ משתנה סטטי שיכיל את מספר התאים התפוסים ע"י צוללות (לדוגמא אם נבנו 3 צוללות באורכים של 5, 4, 3 – המשתנה יכיל את הערך 12 המציין כי - 12 תאים מתוך לוח המשחק תפוסים ע"י צוללות).

מתודות המחלקה :

- בנאי המקבל את מספר הצוללות הרצוי (חייב להיות בין 1 ל-5, אחרת תבנה צוללת אחת בלבד)
- ההורס המחשיר זיכרון הדינאמי התפוס ע"י מערך הצוללות
- מתודה פרטית ClearBoard לאיפוס לוח המשחק (איפוס המערך, מאתחל את כל התאים בתו ' _').
- מתודה פרטית CheckPlace שתבדוק שאכן ניתן להציב את הצוללת במקום שנבחר. אם לא, יש למצוא עבורה מקום אחר. לדוגמא, בלוח של 10*10 נבחר להציב במאונך צוללת באורך 5 במיקום שורה 8 עמודה A. ניתן לראות כי, אין לצוללת מקום כי מ-8 עד-10 יש רק 3 מקומות ולכן יש לנסות להציב אותה שנית.
- מתודה פרטית BuildBoard להצבת הצוללות (יש להתייחס לכללים, יוסבר בהמשך). הצבת הצוללת במאונך או במאוזן תקבע באופן אקראי. מיקום הצוללת יקבע באופן אקראי.
- מתודה PrintBoard להדפסת לוח המשחק (יש להדפיס מעין טבלה כאשר החלק העליון הוא אותיות והצד השמאלי מספרים. בשלב הראשון הלוח ריק. כל ניחוש אמור לשנות את הלוח כך שתופיע תוצאת הניחוש למשל @ עבור פגיעה ו- X עבור החטאה.
- מתודה PrintOriginalBoard להדפסת לוח המשחק עם הצוללות המוצבות (כאילו שאנחנו "מרמים" ורואים הכל) הצוללות יסומנו ע"י ה S .

- מתודה פרטית CheckHit שבודקת, אם המקום שנבחר פגע בצוללת. אם כן יש להודיע על פגיעה ולסמן את הפגיעה בלוח (ע"י הצבת @ במקום המתאים).
אם לא יש לסמן שהמקום נבחר ואין בו צוללת (נניח ע"י הצבת X במקום המתאים). אם כבר בחרנו בעבר במקום יש לתת הודעה על כך.
- מתודה פרטית CheckSunk מקבלת 2 קואורדינטות שבהן נפגעה הצוללת ובודקת אם כל הצוללת הוטבעה.
- מתודה Play שמנהלת את המשחק – יש לתת אפשרות בסיבוב הראשון לראות איך המחשב סידר את הצוללות ולשחק משם. הפונקציה תבדוק אם התגלו כל הצוללות והמשחק נגמר.
- הצבת הצוללות נעשית לפני תחילת המשחק. ישנם כמה חוקים לפיהם מציבים את הצוללות (הכללים):
 - יש להציב את הצוללות במאונך או במאוזן, ולא באלכסון.
 - אין לחפוף את הצוללות (כלומר, שאחת תהיה מוצבת על השנייה).
 - אין להצמיד את הצוללות אחת לשנייה. גם לא באלכסון.



חלק שלישי :

- יש לכתוב תוכנית ראשית (Driver) אשר יוצרת משחק חדש ומאפשרת לשחקן לגלות את הצוללות.
- בשלב ראשון יש לשאול את המשתתף כמה צוללות הוא רוצה על הלוח הבחירה (בין 1 ל-5).
- לאחר מכן (אך לפני שהמשחק מתחיל) יש לאפשר למשתמש (אם הוא רוצה) לראות את הלוח המקורי.

איור מס' 4:

```
C:\Windows\system32\cmd.exe
How many battleships do you want to place on board?
4
Do you want a hint? [y/n]
y
  A B C D E F G H I J
1  - - - - - - - - -
2  - - - - - S - - S
3  - S S - - - S - - S
4  - - - - - S - - S
5  - - - - - - - - S
6  - - - - - - - - -
7  - - - - - - - - -
8  - - - - - - - - -
9  - - - S S S S S - -
10 - - - - - - - - -
Memorize the board and press enter to start the game.._
```

- משלב זה ניהול המשחק יבוצע על ידי לוח המשחק (מחלקה Game).
איור מס' 5:

```
C:\Windows\system32\cmd.exe
Hit!!!
Battleship Sunk!!!
  A B C D E F G H I J
1  - - - - - - - - -
2  - - - - - - - - -
3  - - - - - - - - -
4  - - - - - - - - -
5  - - - - - - - - -
6  - - - - - - - - -
7  - - - - - - - - -
8  - - - - - - - - -
9  - - X @ @ @ @ @ - -
10 - - - - - - - - -
Enter guess: x y <to exit use 0 0>
b 3_
```

- בתום המשחק יש להראות את מצב הלוח הסופי.

איור מס' 6:

```
C:\Windows\system32\cmd.exe
Hit!!!
Battleship Sunk!!!
WIN!!!
You had 17 rounds total.
  A  B  C  D  E  F  G  H  I  J
1  _  _  _  _  _  _  _  _  _
2  _  _  _  _  _  _  e  _  _  e
3  _  e  e  x  _  _  e  _  _  e
4  _  _  _  _  _  _  e  _  _  e
5  _  _  _  _  _  _  _  _  _  e
6  _  _  _  _  _  _  _  _  _  _
7  _  _  _  _  _  _  _  _  _  _
8  _  _  _  _  _  _  _  _  _  _
9  _  _  x  e  e  e  e  e  _  _
10 _  _  _  _  _  _  _  _  _  _
Press any key to continue . . . _
```

עבודה פוריה !!!