מבוא למדעי המחשב, סמסטר קיץ 2018

תרגיל בית 2

מועד אחרון להגשה: **יום ב' – 20.8.2017 23:59**

המתרגל האחראי על תרגיל זה: **עידו רפאל**

משרד: טאוב 328
E-mail: ungulient@gmail.com (נא להוסיף את מספר הקורס –"234114" לכותרת המייל, כדי שהמייל יגיע לתיקייה הנכונה ולא אפספס אתכם)
שעת קבלה רגילה: יום ב' 15:30-16:30 (בתיאום מראש במייל או לאחר התרגול).

הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרונן.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל למתרגל האחראי על תרגיל זה עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם אתר הבדיקה ועם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
  - השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
  - וודאו את האותיות הגדולות והקטנות לפי הדוגמאות וההסברים בתרגיל.
  - אין להדפיס רווחים שלא התבקשתם להדפיס (בתחילת שורה או בסופה).
- בתרגיל זה מותר להשתמש בפונקציות מהספרייה stdbool.h, stdlib.h, stdio.h למעט במקרים בהם נאמר אחרת. החומר הנדרש לתרגיל זה שייך לתרגולים 1-5. אין להשתמש בחומר שאינו מופיע במצגות אלה.
- ההגשה הינה אלקטרונית ו**בבודדים** דרך אתר הקורס. קובץ ההגשה יהיה מסוג **zip** (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
  - קובץ **students.txt** עם שמך **באנגלית**, מספר תעודת הזהות וכתובת האי-מייל שלך.
  - קובץ פתרון **hw2q1.c** עבור שאלה 1.
  - קובץ פתרון **hw2q2.c** עבור שאלה 2.
  - קובץ פתרון **hw2q3.c** עבור שאלה 3.
- **חובה לשמור את קוד אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!** אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

בדיקה ידנית - בונוס (10 נקודות לציון התרגיל):

בנוסף לבדיקה האוטומטית, התרגיל ייבדק גם בבדיקה ידנית. הבדיקה תתמקד בנושאים הבאים:

- אורך **מימוש** (לא כולל ההכרזה, או שם הפונקציה) כל פונקציה לא יעלה על 16 שורות קוד.

  הגבלה זו תקפה לכל הפונקציות, כולל main.

  רק שורות ריקות, שורות עם סוגר מסולסל **בלבד**, ושורות עם הערות **בלבד לא** נספרות.

  רווחים לבנים לא נספרים.

  אסור לכתוב כמה פקודות שונות משמעותית באותה השורה, למשל כותרת תנאי\לולאה

  צריכה להיות בשורה נפרדת מגוף התנאי\לולאה.

- רוחב שורה - רוחב כל שורה (**כולל הערות והזחות**) לא יעלה על **80** תווים.

ניתן לסמן אורך של שורה בקודבלוקס:

Code::Blocks -> Settings -> Editor -> Margins and caret ->

Right margin hint: Visible line, **Hint column**: 80

- קבועים בקוד:
  - יש להגדיר בעזרת #define כל קבוע משמעותי. שמות קבועים צריכים להיות באותיות גדולות בלבד, אם השם מכיל יותר ממילה אחת מילים יופרדו בעזרת מקף תחתון (למשל STUDENTS_NUM).
  - אין להשתמש בערכי ASCII ישירות. יש להשתמש בייצוג של התווים (למשל 'a').
- הזחות:
  - שיטת הזחה מקובלת – הזחת קוד בכל בלוק, למשל:

```
int main()
{
	// your code here
	while (…)
	{
		// your code here
	}
}
```

- אין להשתמש במשתנים גלובאליים או סטטיים.
- שמות משתנים/קבועים/פונקציות צריכים להיות אינפורמטיביים, להעיד על מטרתם.
- בהירות הקוד ותיעוד:
  - יש לתעד את הקוד באמצעות הערות באנגלית בלבד. במידה ויש כמה שורות קוד שניתן להסביר בקצרה מה המטרה שלהן, יש לשים הערה בהתחלה ואין צורך לתעד כל שורה.
  - יש לתעד פונקציות – לפני הפונקציה להוסיף הערה שמסבירה בקצרה מה הפונקציה עושה ומה המשמעות של הפרמטרים שלה.
  - התיעוד צריך להיות אינפורמטיבי, כלומר יש להסביר מה המטרה של שורות הקוד ולא לכתוב את הקוד במילים.
- שכפול קוד שלא לצורך (למשל ריבוי קוד זהה במספר מקרי if-else שונים).
- אי-עמידה באחת מדרישות התרגיל (שימוש בחומר שהיה אסור בתרגיל וכו').

באופן כללי – הקפידו על כתיבת קוד מסודר ומובן ככל שניתן תוך יישום העקרונות שנלמדו בכיתה.
מותר לכם לממש פונקציות עזר משלכם ולהשתמש בהן.

כלומר, המתואר לעיל אינו חובה, אך כדי לקבל את הבונוס צריך לעמוד בדרישות.

אם עומדים בדרישה הנ"ל – אורך כל פונקציה לא יעלה על 16 שורות קוד. הגבלה זו תקפה לכל הפונקציות, כולל main. רק שורות ריקות, שורות עם סוגר מסולסל בלבד, ושורות עם הערות בלבד לא נספרות. אסור לכתוב כמה פקודות שונות משמעותית באותה השורה, למשל כותרת תנאי\לולאה צריכה להיות בשורה נפרדת מגוף התנאי\לולאה.
אז ניתן לקבל בונוס של 10 נקודות.

שימו לב:

באתר הקורס פורסמו 3 קבצי שלד לשלושת השאלות בתרגיל תחת השמות:

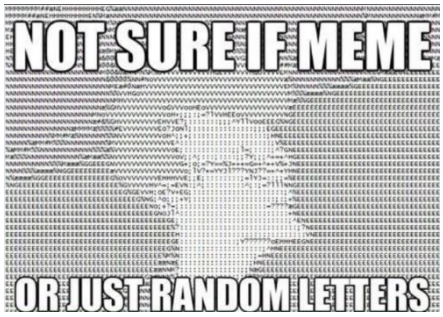
- **hw2q1.c**
- **hw2q2.c**
- **hw2q3.c**

בקבצים אלו קיימות כבר פונקציות הדפסה (שהשמות שלהן מתחילים ב-print), ובאמצעותם תדפיס התוכנית שלכם פלט כפי שיפורט בהמשך בכל שאלה.

אתם יוצרים את הפרויקטים שלכם לכל שאלה (לדוגמא פרויקט בשם **hw2q1** כפי שנלמד בעבר, ואיך שעשיתם בתרגילים הקודמים, עבור שאלה 1), אך תחליפו את קובץ ה-c שיווצר בפרויקט, בקובץ המתאים לשאלה זו שפורסם באתר לדוגמא **hw2q1.c** עבור פרויקט של שאלה 1, ולשם תמשיכו להוסיף את הקוד שלכם לפתרון השאלות, בין אם זה קוד בתוך ה-main או מימוש פונקציות נוספות מחוץ ל-main.

אלו הם גם שלושת הקבצים שתגישו יחד עם קובץ **students.txt**.

בכל שאלה בתרגיל אתם יכולים להוסיף פונקציות עזר כרצונכם.



שאלה 1:

בשאלה זו עליכם לכתוב תוכנית המקבלת שורה של תווים מהמשתמש ומדפיסה עבור כל תו את הנתונים הבאים:

- התו עצמו.
- ערך ה-ASCII של התו.
- ריבוע ערך ה-ASCII של התו.
- הפרש ערך ה-ASCII של התו מערך ה-ASCII של התו הקודם בסדרה (עבור התו הראשון שדה זה יכיל את ערך ה-ASCII של התו עצמו).
  הערה: אם ההפרש חיובי אין צורך להוסיף לפניו את הסימן "+". אם שלילי אז מודפס המספר עם הסימן "-" (או בקיצור, הדפסה רגילה).
- ספרת האחדות של **ההפרש**, כולל הסימן רק במקרה שהוא שלילי (שוב הדפסה רגילה).

מבצעים את ההדפסה עבור כל תו דרך הקריאה לפונקציה printGivenParamsToTheOutput עם העברת הפרמטרים הנכונים, לאחר שחושבו לפי הסדר לעיל.

בנוסף, לבסוף יש להדפיס את הסטטיסטיקות הבאות:

- כמות הספרות (תווים 0-9) שהתקבלו.
- כמות האותיות (תווים a-z ו-A-Z) שהתקבלו.
- כמות הרווחים שהתקבלו (**רק** רווחים "רגילים", כלומר, התו שערך ה-ASCII שלו הוא 32. **לא** טאבים, **לא** ירידות שורה, או כל תו אחר נחשב).

את הדפסה זו מבצעים דרך קריאה לפונקציה printResults עם העברת הפרמטרים הנכונים לאחר שחושבו לפי הסדר לעיל.

התוכנית תסיים את ריצתה כאשר המשתמש ידפיס את התו ';' (נקודה פסיק). שורת המידע לא תודפס עבור תו זה.

ניתן להניח כי <u>לא יתקבל</u> EOF לפני התו ';'. אין הגבלה על אורך הקלט מהמשתמש.

אתגר רק לשאלה 1 (ללא בונוס):

מי שמחפש אתגר (כלומר, חלק זה **אינו** חובה, ותקף רק לשאלה **1**), מוזמן לנסות מימוש שלא מכיל שימוש במשפטי תנאי:

- If-else
- Switch
- האופרטור הטרנארי :?
- שימוש בלולאות כדי לדמות משפטי תנאי (כמו while על התנאי, ואז break). כמובן ששימוש **רגיל** בלולאות מותר, וגם שימוש ב-break מותר.
  דוגמא למה אסור:

```
while (x>0) {
        x++;
        break;
}
```

שזה בעצם כמו לכתוב if (x>0) x++;.

רמז לאתגר: בתרגול 3 פתרנו את תרגיל 2 בשתי דרכים. בדרך השניה עשינו שימוש בכוחם של משתנים בוליאנים, ולא היינו צריכים משפטי תנאי. נסו קודם לפתור רגיל, ואח"כ להחליף את משפטי התנאי בעזרת משתנים בוליאנים.

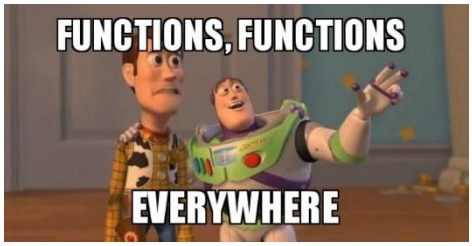
הערה: כמובן שבקוד אמיתי (לא של שיעורי בית) שימוש במשפטי תנאי הוא הרבה יותר **קריא** ונכון, אבל עדיין כדאי לדעת גם לשחק עם משתנים בוליאנים.

דוגמאות הרצה:

(לדוגמאות נוספות, בדקו את קבצי הקלט והפלט שצורפו לתרגיל)

```
C:\progs\hw2>hw2q1
a0 {8]zCv @qQ;
a         97      9409        97         7
0         48      2304       -49        -9
          32      1024       -16        -6
{        123     15129        91         1
8         56      3136       -67        -7
]         93      8649        37         7
z        122     14884        29         9
C         67      4489       -55        -5
v        118     13924        51         1
          32      1024       -86        -6
@         64      4096        32         2
q        113     12769        49         9
Q         81      6561       -32        -2
Number of digits received: 2
Number of letters received: 6
Number of spaces received: 2
```

```
C:\progs\hw2>hw2q1
abcsA987gg";
a         97      9409        97         7
b         98      9604         1         1
c         99      9801         1         1
s        115     13225        16         6
A         65      4225       -50         0
9         57      3249        -8        -8
8         56      3136        -1        -1
7         55      3025        -1        -1
g        103     10609        48         8
g        103     10609         0         0
"         34      1156       -69        -9
Number of digits received: 3
Number of letters received: 7
Number of spaces received: 0
```



שאלה 2: מערכים חד מימדיים ופונקציות

ממשו את הפונקציות הבאות:

שימו לב כי מרגע הגדרת הפונקציות, ניתן להשתמש בהן בכל מקום בתרגיל. כאשר נאמר כי "הפונקציה מקבלת" הכוונה לפרמטרים, ולא לקלט מן המשתמש.

1. פונקציה בשם **average** המקבלת שלושה מספרים שלמים, ומחזירה את הממוצע שלהם בקיצוץ הספרות שאחרי הנקודה (למשל אם הממוצע היה צריך להיות 3.8 הוא יהיה 3 ואם הוא היה אמור להיות 7.9- הוא יהיה 7-).
2. פונקציה בשם **min** המקבלת שני מספרים שלמים, ומחזירה את המספר המינימלי מביניהם.
3. פונקציה בשם **absolute** המקבלת מספר שלם ומחזירה את ערכו המוחלט.
4. פונקציה בשם **root** המקבלת מספר שלם, ומחזירה את השורש שלו במידה והשורש הוא מספר שלם, אחרת מחזירה 1-.
   שימו לב כי יש לוודא שהמספר **לא** שלילי, אחרת (אם הוא שלילי) הפונקציה מחזירה 1-.
5. פונקציה בשם **power** המקבלת שני מספרים שלמים (נקרא להם בסיס ומעריך) ומחזירה את הבסיס בחזקת המעריך. אם התוצאה גדולה ממש מ-10^9 (מיליארד. כלומר 9 אפסים), או אם אחד הקלטים שלילי, יש להחזיר 1-.
6. פונקציה בשם **characterAnalysis** שמדפיסה בהתחלה הודעת פתיחה מתאימה דרך קריאה לפונקציה printCharactersOpenMessage ואז **קולטת מהמשתמש** סדרה של תווים עד שמתקבל EOF.
   עבור כל אות קטנה (a-z) מדפיסה כמה פעמים הופיע בסדרה שהתקבלה, ע"י קריאה לפונקציה printDataPerGivenCharAsInput עם העברת הפרמטרים הנכונים לפונקציה (תו ומספר הפעמים שהופיע – ראו את מימוש הפונקציה בקוד).
   לבסוף תדפיס כמה תווים הופיעו ש**אינם** מהתחום (a-z) כולל רווחים, **אך ללא ספירת ירידות שורה (התווים '\n' ו-'\r')** ע"י קריאה לפונקציה printNumberOfOtherCharsAppeared עם העברת הפרמטר הנכון לפונקציה.
   שימו לב שתווים מהתחום (A-Z, כלומר אותיות גדולות) אינם נחשבים לתווים מהתחום (a-z).

ניתן להניח בתרגיל זה כי 0 בחזקת 0 שווה ל-1, למרות שערך ביטוי זה לא מוגדר.

כתבו תכנית אשר:

1. מדפיסה את הודעת הפתיחה של שלושה מספרים דרך קריאה לפונקציה printThreeDigitsOpenMessage
2. קולטת שלושה מספרים שלמים (במידה והקלט אינו 3 מספרים שלמים, כלומר לא תקין, יש **לצאת** מהתכנית).
3. מבצעת (ההדפסות יתבצעו ע"י הפונקציה printResults):
   a. **הדפסה** של ממוצע המספרים בקיצוץ הספרות שאחרי הנקודה (כפי שתואר ב-**average**).
   b. מצאו את המספר המינימלי מבין השלושה, חשבו את ערכו המוחלט, **והדפיסו** את שורשו השלם במידה וקיים (אם לא קיים יש להדפיס 1-).
   c. **מדפיסה** את ממוצע המספרים (בקיצוץ הספרות לאחר הנקודה, כפי שתואר) **בחזקת** המספר המינימלי, ב**ערכו המוחלט**. אם הערך גדול ממש מ-10^9 (מיליארד) או ממוצע המספרים שלילי, יש להדפיס 1- (אם התיאור לא ברור, תסתכלו בדוגמאות הרצה וקלטים לדוגמא).
4. תפעיל את הפונקציה **characterAnalysis** כפי שתוארה בסעיף 6 של מימוש הפונקציות.

שימו לב: אין להשתמש בספריית math.h או כל מימוש מוכן אחר של sqrt או pow. אתם נדרשים לתת את המימוש שלכם. המימוש לא צריך להיות יעיל, ואמור להיות פשוט וברור (המימוש ה"טריוויאלי"). **מותר להשתמש רק בספריות stdio.h, stdlib.h, stdbool.h.**

רמז:

בחישוב power אין צורך לנסות לתת פתרון יעיל, ואפשר להסתפק בפתרון הטריוויאלי - לכפול את הבסיס בעצמו כמספר הפעמים שניתן במעריך, ע"י לולאה.

כדי לבדוק האם תוצאת **הכפל** a * b (כאשר a,b הם שני int חיוביים) תחרוג מהמספר NUM (למשל בתרגיל זה נרצה להגדיר את NUM כ-10^9), בודקים האם **(a > NUM / b)** (במידה וכן, תוצאת הכפל תגרום לחריגה מעבר ל-NUM, ולכן נחזיר 1- כפי שתואר).

לכן כאשר נחשב חזקה באופן טריוויאלי, בכל פעם נכפול בבסיס, אך **לפני** שנבצע את הכפל, נבדוק **לפני הכפל** את ה**תנאי**, ובמידה ומתקיים נדע שלאחר הכפל נחרוג, ולכן נחזיר 1-. אחרת התנאי לא מתקיים ואפשר להמשיך לאיטרציה הבאה).

שימו לב שהתנאי מבצע חלוקה ב-b, לכן כדאי לדאוג למקרה ש-b הוא 0 לפני, בנפרד.

למי שמתעניין למה, קראו את התגובה כאן https://stackoverflow.com/a/1514309 (שימו לב שמדובר כאן על חריגה מעל גבולות ייצוג של int, וכאן בתרגיל אנחנו מדברים על חריגה מעל 10^9).

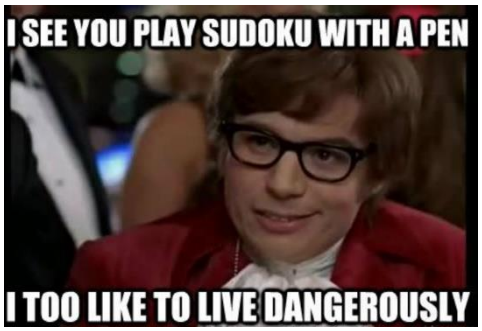
נסו ליצור לעצמכם טסטים בהם תוצאת החזקה אכן חורגת, וודאו שהתכנית שלכם תופסת מקרים אלה, ונותנת פלט בהתאם.

דוגמאות הרצה:

(לדוגמאות נוספות, בדקו את קבצי הקלט והפלט שצורפו לתרגיל)

```
Please enter three integers:
10 5 -9
2 3 512
Please enter your character set:
aaa bbb zzz AAA
^Z
letter a appeared 3 time in the given set of characters
letter b appeared 3 time in the given set of characters
letter c appeared 0 time in the given set of characters
letter d appeared 0 time in the given set of characters
letter e appeared 0 time in the given set of characters
letter f appeared 0 time in the given set of characters
letter g appeared 0 time in the given set of characters
letter h appeared 0 time in the given set of characters
letter i appeared 0 time in the given set of characters
letter j appeared 0 time in the given set of characters
letter k appeared 0 time in the given set of characters
letter l appeared 0 time in the given set of characters
letter m appeared 0 time in the given set of characters
letter n appeared 0 time in the given set of characters
letter o appeared 0 time in the given set of characters
letter p appeared 0 time in the given set of characters
letter q appeared 0 time in the given set of characters
letter r appeared 0 time in the given set of characters
letter s appeared 0 time in the given set of characters
letter t appeared 0 time in the given set of characters
letter u appeared 0 time in the given set of characters
letter v appeared 0 time in the given set of characters
letter w appeared 0 time in the given set of characters
letter x appeared 0 time in the given set of characters
letter y appeared 0 time in the given set of characters
letter z appeared 3 time in the given set of characters
number of other characters in the given set was: 6
```

```
Please enter three integers:
-3 -4 -8
-5 -1 -1
Please enter your character set:
abcde!@#%^&*()
^Z
letter a appeared 1 time in the given set of characters
letter b appeared 1 time in the given set of characters
letter c appeared 1 time in the given set of characters
letter d appeared 1 time in the given set of characters
letter e appeared 1 time in the given set of characters
letter f appeared 0 time in the given set of characters
letter g appeared 0 time in the given set of characters
letter h appeared 0 time in the given set of characters
letter i appeared 0 time in the given set of characters
letter j appeared 0 time in the given set of characters
letter k appeared 0 time in the given set of characters
letter l appeared 0 time in the given set of characters
letter m appeared 0 time in the given set of characters
letter n appeared 0 time in the given set of characters
letter o appeared 0 time in the given set of characters
letter p appeared 0 time in the given set of characters
letter q appeared 0 time in the given set of characters
letter r appeared 0 time in the given set of characters
letter s appeared 0 time in the given set of characters
letter t appeared 0 time in the given set of characters
letter u appeared 0 time in the given set of characters
letter v appeared 0 time in the given set of characters
letter w appeared 0 time in the given set of characters
letter x appeared 0 time in the given set of characters
letter y appeared 0 time in the given set of characters
letter z appeared 0 time in the given set of characters
number of other characters in the given set was: 9
```



שאלה 3: בודק פתרונות סודוקו

סודוקו הוא תשבץ מספרים שבו צריך למקם את המספרים 1 עד n על לוח משובץ שגודלו nXn המחולק ל-n ריבועים בגודל $\sqrt{n}\ X\ \sqrt{n}$ כל אחד.

מטרת המשחק – למקם את n המספרים על גבי לוח המשחק כך שבכל טור, בכל שורה, ובכל ריבוע, יופיע כל מספר בדיוק פעם אחת.

בשאלה זו נרצה לממש בודק פתרונות לסודוקו. התכנית תקבל את גודל הלוח n ומטריצה בגודל nXn ותבדוק האם המטריצה הנתונה מהווה פתרון חוקי לסודוקו.

דוגמא לפתרון חוקי עבור n=9 (הפתרון חוקי היות ובכל שורה, טור וריבוע מודגש, מופיעים המספרים 1-9 בדיוק פעם אחת).

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

התכנית:

1. תדפיס למשתמש הודעת פתיחה מתאימה להכנסת גודל הלוח, ע"י קריאה לפונקציה printOpenMessageForSudokuSize.
2. תקלוט מהמשתמש את n.
   ניתן להניח כי יוכנס מספר שלם בטווח גודלו של int.
   אך **יש לוודא** שהוא חיובי ממש (כלומר 0 גם אינו חוקי), וכן **יש לוודא** שיש לו שורש שלם.
   לצורך כך תוכלו להעזר בפונקציה שמימשתם בשאלה הקודמת (בשלב זה ניתן פשוט להעתיק את הקוד). שימו לב, **אסור להשתמש ב-sqrt מספריית math.h**.
   במידה והמספר אינו חוקי (כלומר שלילי או 0), יש לקלוט מספר נוסף שוב ושוב עד שמתקבל מספר חוקי.
   ניתן להניח כי ה-n שיתקבל יהיה **קטן\שווה** ל-25, כלומר שגודל המטריצה הוא מקסימום 25X25.
3. לאחר קבלת n חוקי, התכנית תדפיס למשתמש הודעה מתאימה להכנסת הפתרון, ע"י הפונקציה printOpenMessageForSudokuSolution.
4. התכנית תקלוט מהמשתמש את הפתרון, אותו יש לקלוט למערך דו-מימדי בגודל nXn (סדר הכנסת האיברים הוא משמאל לימין, שורה אחר שורה). בשלב זה ניתן להניח שהקלט תקין, רווח מפריד בין המספרים בשורה, ובין כל שורה מפריד ירידת שורה (ניתן לראות דוגמאות בדף הבא), ובפרט כי הקלט מורכב מספרות בין 1 עד n.
5. אם הסודוקו חוקי יש להדפיס הודעה ע"י הפונקציה printValidSolution ואחרת (הפתרון אינו חוקי) יש להדפיס ע"י הפונקציה printBadSolution.

דוגמאות הרצה:

(לדוגמאות נוספות, בדקו את קבצי הקלט והפלט שצורפו לתרגיל)

```
Please enter the size of your soduko:
4
Please enter your solution:
1 2 3 4
3 4 1 2
4 3 2 1
2 1 4 3
Valid solution!
```

גודל שלילי

למשל 1 מופיע פעמיים בריבוע המסומן

```
Please enter the size of your soduko:
-5
4
Please enter your solution:
1 2 3 4
2 1 4 3
3 4 1 2
4 3 2 1
Bad solution!
```

גודל שאין לו שורש שלם

למשל 2 מופיע פעמיים בשורה השנייה

```
Please enter the size of your soduko:
5
4
Please enter your solution:
4 3 2 1
1 2 3 2
3 1 4 3
2 4 1 4
Bad solution!
```