

סמסטר ב', תשפ"ב, מרץ 2022

# עבודת בית 2

## לולאות ופונקציות

מבוא למדעי המחשב

המחלקה להנדסת תוכנה

המכללה האקדמית להנדסה ע"ש סמי שמעון

# הנחיות

מועד הגשה: 27.03.2022

מועד פרסום: 14.03.2022

מתרגלת אחראית: הד"ר כוכבי

## מועד הגשה

1. ההגשה היא עד סוף יום ההגשה, כלומר עד השעה 23:59 באותו היום. אל תחכו לרגע האחרון. תכננו את זמנכם בהתאם. הגישו לפני.
2. איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו. בכל מקרה לא יהיה ניתן להגיש מעבר ל-2 ימי איחור ממועד ההגשה דלעיל.
3. במקרים חריגים בלבד ניתן לפנות למרצה לאישור הגשה באיחור מעבר ליומיים הנ"ל.

## אופן הגשה

4. קראו היטב את השאלות. עליכם לענות על כל השאלות בעבודה זו.
5. הגשת העבודה תהיה דרך אתר הקורס במודל בלבד. הגשת העבודה היא ביחידים.
6. כיצד להגיש?
- a. עבור כל משימה יש ליצור קובץ C עם סיומת c. חובה לתת את השמות הבאים לקבצים ex1.c, ex2.c, ex3.c, ex4.c, ex5.c, ex6.c, ex7.c.
- b. חובה לכווץ את כל הקבצים יחדיו לקובץ אחד ויחיד בפורמט RAR או ZIP, ולהגיש רק אותו.
- c. יש להגיש רק תוכניות שעוברות קומפילציה תקינה ללא שגיאות על קומפיילר שפת Visual Studio 2019 C.
7. בתחילת כל קובץ יש להוסיף את התיעוד הבא. יש לשנות את השם לשם שלכם ואת תעודת הזהות לתעודת הזהות שלכם.

// Assignment: 2

// Author: Israel Israeli, ID: 01234567

## שאלות

8. שאלות לגבי העבודה יש לשאול בפורום באתר המודל של הקורס או בשעות קבלה של המתרגלת/ת האחראית בלבד. אין לשלוח שאלות במייל לא למתרגלת האחראית ולא למתרגלים/מרגלים אחרים.
9. ניתן לשאול שאלות הבהרה ומיקוד על המשימות שבעבודה במידה ומשימה מסוימת לא ברורה. לא ניתן לשאול על הפתרונות שלכם. לדוגמא, לא ניתן לשאול האם הפתרון שלי נכון, לא ניתן לשאול למה הפתרון לא עובד, וכדומה.

## קוד

10. ניתן להשתמש בכל החומר שלמדנו ובכל החומר שנלמד עד "פונקציות" (כולל). בפרט, ניתן להשתמש בבקרת זרימה, לולאות, לולאות מקוננות, פונקציות, וכדומה. אין להשתמש בחומר שנלמד אחרי פונקציות בהרצאות (למשל, אין להשתמש במערכים).
11. ניתן להשתמש בספריית stdio.h, בספריית assert.h ובספריית stdbool.h בלבד. אסור להשתמש בספריות אחרות.
12. הדבר החשוב ביותר הוא שהקוד שלכם יעבוד באופן תקין / נכון. יש לנסות ולייעל את הקוד במקומות בהם ישנה אפשרות להתייעלות.
13. יש להקפיד על תכנות נכון. הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: define, const או enum, בהתאם לצורך. יש לכתוב **הערות באנגלית בלבד**, יש להקפיד על הזחות, כיתוב נכון/קריא, שמות משמעותיים וכיוצא באלו.

## שונות

14. **המשימות בעבודה זו הינן שוות משקל.** כלומר, משקל כל משימה הוא 100 חלקי מספר המשימות בעבודה.
15. בכל המשימות בעבודה זו הניחו כי הקלט מקיים את כל ההנחות הכתובות במשימה. כלומר, **אינכם נדרשים לבדוק שהקלט מקיים את ההנחות** – בין אם מדובר כארגומנט/פרמטר לפונקציה ובין אם מדובר בקלט מהמשתמש.
16. באפשרותכם לכתוב ולהשתמש בפונקציות עזר, במידת הצורך.

**בהצלחה!**

# עבודת בית 2

## לולאות ופונקציות

### משימה 1: מציאת שורש ריבועי

**שורש ריבועי** של מספר  $n$  כלשהו הוא מספר, שאם מכפילים אותו בעצמו מקבלים את  $n$ . הפעולה החישובית של מציאת השורש הריבועי נקראת הוצאת שורש ריבועי.

כתבו פונקציה בשם `squareRoot` המקבלת כקלט מספר שלם אי-שלילי  $n \geq 0$ . **יש להניח כי  $n \geq 0$  ושיש לו שורש ריבועי שהוא מספר שלם.** הפונקציה מחזירה כפלט את השורש הריבועי של  $n$ .

בטבלה הבאה ניתן לראות דוגמאות של הפעלה של הפונקציה ואת התוצאה שאותה היא אמורה להחזיר.

input / application	output
<code>squareRoot(16)</code>	4
<code>squareRoot(64)</code>	8
<code>squareRoot(81)</code>	9
<code>squareRoot(121)</code>	11

## משימה 2: הדפסת השורשים של נוסחא ריבועית

### הגדרה 1

מציאת שורשי המשוואה הריבועית  $ax^2 + bx + c = 0$  מתבצעת ע"י שימוש בנוסחת השורשים:

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

### הגדרה 2

ניתן לדעת את כמות הפתרונות הקיימים למשוואה ריבועית לפי ערך ה  $\Delta$  של נוסחת השורשים:

$$\Delta = b^2 - 4ac$$

אם  $\Delta > 0$  אז למשוואה 2 פתרונות.

אם  $\Delta = 0$  אז למשוואה פתרון יחיד.

אם  $\Delta < 0$  אז למשוואה אין פתרונות.

השורשים  $x_1, x_2$  הם פתרונות המשוואה.

### משימה

כתבו פונקציה בשם **equationSolutions** המקבלת את 3 ערכים  $a, b, c$  אשר יכולים להיות עשרוניים ושליליים (float) ומחזירה את מספר הפתרונות הקיימים עבור המשוואה:

$$ax^2 + bx + c = 0$$

#### דוגמא:

עבור  $a = 16, b = -8, c = 2$  הפונקציה תחזיר 0.

### משימה

כתבו פונקציה בשם **printSquareRoot** שאינה מקבלת קלט ואינה מחזירה (return) ערך. על הפונקציה לבצע את הפעולות הבאות בצורה איטרטיבית עד אשר המשתמש מכריז כי רוצה לסיים:

1. שואלת את המשתמש האם ברצונו להמשיך בפעולות, וקולטת ממנו תו.
2. אם המשתמש הכניס 'y', המשתמש רוצה להמשיך.
3. אחרת, אם המשתמש הכניס תו שונה מ 'y', תדפיס "goodbye" ותסיים.
4. מדפיסה בקשה לקבלת פרמטרי המשוואה הריבועית.
5. מקבלת מהמשתמש 3 ערכים  $a, b, c$ . ניתן להניח כי המשתמש יכול להכניס מספרים עשרוניים ושליליים.
6. בודקת את מספר הפתרונות שיש למשוואה, אם למשוואה פתרון אחד לפחות, תדפיס אותם לפי נוסחת השורשים. אחרת, תדפיס כי למשוואה אין פתרון.

**הערה:** במשימה זו, ניתן להשתמש בפונקציה sqrt השייכת לספרייה math.h.

**דוגמא:**

עבור הערכים  $a = 16$ ,  $b = -8$ ,  $c = 2$  אשר נקלטו מהמשתמש, הפונקציה תדפיס:

"The quadratic equation has no solution"

מכיוון ש  $\Delta = -64$  ולכן למשוואה זו אין פתרונות.

**דוגמא:**

עבור הערכים  $a = 1$ ,  $b = 4$ ,  $c = -17$  אשר נקלטו מהמשתמש, הפונקציה תדפיס:

" $x_1 = 2.58$ ,  $x_2 = -6.58$ "

**דוגמא:**

עבור הערכים  $a = -8$ ,  $b = 8$ ,  $c = -2$  אשר נקלטו מהמשתמש, הפונקציה תדפיס:

" $x_1 = 0.5$ "

למשוואה שורש יחיד מכיוון ש  $\Delta = 0$

## משימה 3: משחק הניחושים

במשימה זו נכין משחק של המשתמש נגד יריב. המשתמש צריך לנחש מספר שיריבו בחר ואם ניחש נכון, זכה. אחרת, היריב ניצח. למשחק זה יכולים להיות כמה סיבובים לפי רצון המשתמש, כאשר בתחילת כל סיבוב תינתן לו האפשרות לבחור האם לסיים את המשחק. בכל סיבוב המשתמש צובר 5 נקודות במידה והצליח לנחש את המספר ולבסוף יקבל את הניקוד הכולל של כל הסיבובים.

כתבו פונקציה בשם **guessYourNumber** המקבלת כקלט 2 מספרים שלמים (int) בני 3 ספרות המספר הראשון יהיה a והמספר השני b. הניחו כי המספרים האלו בני 3 ספרות. הפונקציה מחזירה את הניקוד שהמשתמש צבר מכל הסיבובים.

על הפונקציה לבצע את הפעולות הבאות בצורה איטרטיבית עד אשר המשתמש מכריז כי רוצה לסיים:

1. שואלת את המשתמש האם רוצה לשחק בסיבוב זה וקולטת ממנו תו.
2. אם המשתמש הכניס 'y', המשתמש רוצה להמשיך.
3. אחרת, אם המשתמש הכניס תו שונה מ 'y', תחזיר את סכום הנקודות שצבר מכל סיבובי המשחק.
4. מקבלת מהיריב (באמצעות פונקציית קלט) מספר שלם בעל 3 ספרות בטווח בין a ל-b. זהו המספר שהמשתמש צריך לנחש.
5. במידה והמספר הנבחר אינו בטווח a ו b יש לקלוט את המספר עד אשר יהיה בטווח.
6. אם המשתמש הצליח לנחש, תודפס לו הודעה "Good job!" ויתווספו לו מסיבוב זה 5 נקודות.
7. אחרת, יינתן לו רמז:

**בשלב זה חובה להשתמש בלולאה.**

כל עוד הוא לא מנחש את המספר הנכון, נגלה לו את הספרה הימנית ביותר (ספרת האחדות). במידה ולא ניחש נכון, נגלה לו את הספרה שלאחריה (ספרת העשרות), ולבסוף את ספרת המאות.

להלן דוגמת של הרצת הפונקציה `guessYourNumber(345, 999)` ואינטראקציה שלה עם המשתמש. בדוגמאות, מה שבצבע **אדום** הוקש/הוקלד ע"י המשתמש/היריב. כל השאר זה הדפסות של הפונקציה. Outputn בסוף זה ערך החזרה של הפונקציה.

Do you want to play? 'y'

Enter the number your rival needs to guess: 310

The number is out of range.

Enter the number your rival needs to guess: 876 //This is the number the user needs to guess.

What is your guess? 564

Wrong, let me give you a hint.

The digit is: 6

What is your guess? 546

Wrong, let me give you a hint

The digit is: 7

What is your guess? 576

Wrong, let me give you a hint

The digit is: 8

What is your guess? 876

Good job!

Do you want to play? 'n'

**Output: 5**



## משימה 4: מספרי ארמסטרונג

**מספר ארמסטרונג** הוא מספר בעל  $n$  ספרות שסכום ספרותיו בחזקת  $n$  (כל ספרה מועלית בחזקת  $n$  ואז סוכמים את כל החזקות) הוא המספר עצמו.

### דוגמאות:

- 153 הוא מספר ארמסטרונג  $153 = 1^3 + 5^3 + 3^3$
  - 1634 הוא מספר ארמסטרונג  $1634 = 1^4 + 6^4 + 3^4 + 4^4$
  - 371 הוא מספר ארמסטרונג  $371 = 3^3 + 7^3 + 1^3$
  - 54748 הוא מספר ארמסטרונג. למספר זה יש 5 ספרות, אנו מעלים כל ספרה שלו בחזקת 5 ומקבלים את המספרים הבאים:  
 $5^5 = 3125$ ,  $4^5 = 1024$ ,  $7^5 = 16807$ ,  $4^5 = 1024$ ,  $8^5 = 32768$   
 נסכום את המספרים שקיבלנו ונראה שהסכום שקיבלנו הוא המספר עצמו: 54748.
  - 8429 הוא **לא** מספר ארמסטרונג מכיוון ש:  $10929 = 8^4 + 4^4 + 2^4 + 9^4$
  - 23723 הוא **לא** מספר ארמסטרונג מכיוון ש:  $17357 = 2^5 + 3^5 + 7^5 + 2^5 + 3^5$
- כתבו פונקציה בשם **printArmstrong** שאינה מקבלת קלט ואינה מחזירה (return) ערך.

על הפונקציה לבצע את הפעולות הבאות:

1. קולטת מהמשתמש מספר שלם וחיובי high.
  2. בודקת כי הערך הנקלט הוא מספר בעל 3 ספרות **לפחות**. אחרת, תמשיך לקלוט עד שיתקבל ערך תקין.
  3. מדפיסה את כל המספרים שהם מספרי ארמסטרונג בטווח מ 0 עד high כולל.
- להלן דוגמאות של הרצת הפונקציה ואינטראקציה שלה עם המשתמש. בדוגמאות, מה שבצבע **אדום** הוקש/הוקלד ע"י המשתמש. כל השאר זה הדפסות של הפונקציה.

### דוגמא:

Enter the upper range number: **50654**  
 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474

### דוגמא:

Enter the upper range number: **100000**  
 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084

### דוגמא:

Enter the upper range number: **999999**  
 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834



## משימה 6: כמות מחלקים מקסימלית

כתבו פונקציה בשם **printMaxCountDivisors** שתקבל שני מספרים שלמים חיוביים low ו-high המהווים תחום.

על הפונקציה לבצע את הפעולות הבאות:

1. מוצאת בתחום את המספר שכמות מחלקיו (מספרים המחלקים אותו ללא שארית) היא המקסימלית לעומת שאר המספרים בטווח.
2. הפונקציה תדפיס את המספר, את כמות מחלקיו, ואת המספרים המחלקים אותו. במידה ויש כמה מספרים בטווח עם כמות מחלקים מקסימלית זהה, נדפיס את המידע עבור כל אחד מהם.

כתבו פונקציה בשם **maxCountDivisors** שאינה מקבלת קלט ואינה מחזירה (return) ערך.

על הפונקציה לבצע את הפעולות הבאות:

1. קולטת מהמשתמש 2 מספרים שלמים חיוביים low ו-high.
2. בודקת אם המספרים אכן חיוביים, במידה ולא, מדפיסה הודעת שגיאה מתאימה ומבקשת לקלוט שנית את הערכים עד שיתקבלו ערכים חיוביים.
3. מוודאה כי low אכן קטן מ-high. במידה ולא, יש לבצע החלפה ביניהם כך ש  $low \leq high$ .
4. הפונקציה תעביר אותם לפונקציה **printMaxCountDivisors**.

להלן דוגמאות הרצה של הפונקציה `maxCountDevisors()` ואינטראקציה שלה עם המשתמש. מה שבבוע **אדום** הוקש/הוקלד ע"י המשתמש. כל השאר זה הדפסות של הפונקציה.

**דוגמא:**

Enter the ranges: **950, 1050**

The number with maximal devisors in the **range of 950 and 1050** is: **1008** it has **30** devisors, and it divides by: **1, 2, 3, 4, 6, 7, 8, 9, 12, 14, ..., 336, 504, 1008.**

**דוגמא:**

Enter the ranges: **33, 20**

The number with maximal devisors in the **range of 20 and 33** is: **24** it has **7** devisors, and it divides by: **2, 3, 4, 6, 8, 12, 24.**

The number with maximal devisors in the **range of 20 and 33** is: **30** it has **7** devisors, and it divides by: **2, 3, 5, 6, 10, 15, 30.**

## משימה 7: בדיקת סדרת תווים

כתבו פונקציה בשם **checkChar** המקבלת תו ובודקת האם התו שקיבלה הוא אות לטינית או תו אחר.  
במידה וקיבלה תו לטיני תחזיר true. אחרת, מחזירה false.

דוגמא:

input / application	output
checkChar('A')	true
checkChar('a')	true
checkChar('\$')	false
checkChar('^')	false
checkChar('t')	true

כתבו פונקציה בשם **checkLetters** אשר אינה מקבלת אף פרמטר.  
על הפונקציה לבצע את הפעולות הבאות:

1. קולטת סידרת תווים מהמקלדת. כאשר התו האחרון הוא רווח או ירידת שורה ('\n').
  2. אם התווים שנקלטו הם אותיות גדולות וקטנות לסירוגין, תחזיר true.
  3. אחרת, תחזיר false.
  4. אם אורך הסידרה שנקלטה הוא 0, הפונקציה תחזיר true.
  5. אם אורך הסידרה שנקלטה הוא 1:
- אם התו הנקלט הוא אות לטינית תחזיר true.
  - אחרת, תחזיר false.

כתוב פונקציה בשם **proofreading** אשר אינה קולטת פרמטרים.  
הפונקציה מפעילה את הפונקציה **checkLetters**. במידה והפונקציה **checkLetters** מחזירה true, הפונקציה **proofreading** תדפיס: "Correct". אחרת, מדפיסה "Incorrect."

להלן דוגמת הרצה של הפונקציה `proofreading()` ואינטראקציה שלה עם המשתמש. מה שבצבע **אדום** הוקש/הוקלד ע"י המשתמש. כל השאר זה הדפסות של הפונקציה.

Enter your string:

Correct.

Enter your string: **y**

Correct.

Enter your string: **\***

Incorrect.

Enter your string: aAaAaAaA

Correct.

Enter your string: AbCdEfG

Correct.

Enter your string: abcDEF

Incorrect.

**בהצלחה!**