

תרגיל 2 – לולאות, לולאות מקוננות ופונקציות

הגשה עד 10/12/19

הוראות הגשה:

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק ל- גב' סבטלנה רוסין, במייל:

sceassign2016@gmail.com

2. תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!

3. להגיש רק תכניות שעוברות קומפילציה על מהדר שפת C gcc Ubuntu Linux.

4. **ההגשה היא של קובץ הקוד (קובץ .c) בלבד - יש לתת לקובץ שם ass2.c**

5. בתחילת הקובץ יש להוסיף את התיעוד הבא:

/* Assignment: 2

Author: Israel Israeli, ID: 01234567

*/

כמוכן שיש לעדכן את השמות ומספרי תעודות זהות שלכם.

6. הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קחבים ומחלה חריפה!) ובצרוף

אישורים מתאימים. כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה **לפחות יומיים**

לפני חלוף הדד-ליין!

7. ההגשה היא עד התאריך האחרון לתרגיל: 10/12/19 בשעה 23:50. הגשה מאוחרת אפילו

בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את

זמנכם בהתאם!

8. **מותר לכם ומומלץ ליצור פונקציות עזר.**

9. אין להשתמש בתרגיל בחומר שטרם נלמד, או שנלמד לאחר משה התרגיל, אלא אם נכתב

במפורש בתרגיל שמותר (אסור להשתמש במערכים, רקורסיה וכו)

10. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום שצריך להכניס מספר שלם – נכניס מספר

שלם (ולא שבר או אות). אנחנו לא מתחייבים שהוא יהיה חיובי או א-שלילי, או בטווח מסוים –

אלא אם כן נאמר אחרת בשאלה עצמה.

11. אחרי כל הדפסה יש לבצע ירידת שורה.

12. בתרגיל יש להשתמש בספריות stdio.h, stdlib.h, time.h בלבד!

13. יש להקפיד על תכנות נכון:

- a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: `define`, `const` או `enum`, בהתאם לצורך.
- b. יש לרשום הערות.
- c. יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!
- d. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן.
- e. לפני בקשת קלט (`scanf`) יש להדפיס למשתמש הוראה (`printf`) איזה קלט מבוקש.
- f. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו').
- g. יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה.

בהצלחה ☺

בתרגיל זה עליכם לממש את כל מה שכתוב בתרגיל. שימו לב כי **מדובר בתוכנית אחת** (!) - פונקציה `main` אחת! חישבו היטב כיצד לחלק את התוכנית לפונקציות (כל פעולה סמנטית היא פונקציה נפרדת!)

המערכת :

בהרצת התוכנית המשתמש יקבל תפריט הבא : **10 נק'**

Menu:

Please choose an option:

- 1 – Check number
- 2 – Prime Palindrome scavenger hunt
- 3 – Remove all min digits
- 4 – The guessing game
- 5 - Print Pattern
- 6 – Exit

- אפשרות (1), תפעיל את פונקציה בשם **CheckAndPrint**.
- אפשרות (2), תפעיל את פונקציה בשם **PrimePalindroms**.
- אפשרות (3), תפעיל את פונקציה בשם **RemoveMinDigit**.
- אפשרות (4), תפעיל את פונקציה בשם **Game**.
- אפשרות (5), תפעיל את פונקציה בשם **PrintPattern**.

הסעיפים הבאים מתייחסים ליישום האופציות בתפריט שהוצג, יש לאפשר למשתמש לבחור בתפריט את אחת האופציות ואז לבצע פעולות הקלט והפלט כפי שנדרש לפי האופציה שבחר. ההרצה תעבוד בצורה כזאת עד לבחירת אופציה האחרונה, כאשר התוכנית תסתיים ותדפיס הודעה הבאה:
Have a nice day !

תרגיל 1 (15 נק')

כתבו פונקציה בשם **IsNumberInRange** המקבלת מספר שלם *number* ושני מספרים שלמים *low* ו-*high* (כלומר $low \leq high$). ומחזירה במקרה הזה (1) **TRUE** אחרת מחזירה (0) **FALSE**.

כתבו פונקציה בשם **SumDigits** המקבלת מספר שלם *number* ומחזירה את סכום ספרותיו.

כתבו פונקציה בשם **PrintNumbers** המקבלת מספר שלם *number*, ומדפיסה את כל המספרים התלת-ספרתיים שסכום ספרותיהם שווה ל-*number*. למטרה זו הפונקציה נעזרת בפונקציה *SumDigits*.

כתב פונקציה בשם **CheckAndPrint** שתקלט שלושה מספרים, low, high, number, ותעביר אותם כפרמטר לפונקציה `IsNumberInRange`. כל עוד הפונקציה `IsNumberInRange` מחזירה `FALSE` (0), הפונקציה `CheckAndPrint` קולטת מהמסך ערכים חדשים וקוראת באמצעותם לפונקציה `IsNumberInRange`. אם הערך המוחזר הוא `TRUE` (1), אז `CheckAndPrint` קוראת לפונקציה `PrintNumbers` עם פרמטר `number` ומדפיסה את התוצאה.

number=5 ,low=0, high=19

הפלט:

104, 113, 122,

תרגיל 2 (15 נק')

פוליןדרום: הוא מספר שניתן לקרוא אותו באותה הצורה הן משמאל לימין והן מימין לשמאל – דוגמאות:

מספר הראשוני: כל מספר השונה מ-1 המתחלק ללא שארית בעצמו וב-1 בלבד – דוגמאות: 2, 17 וכו'

כתבו פונקציה בשם **PrimePalindroms** המקבלת תחום הבנוי משני מספרים חיוביים שלמים, כאשר התחום אינו מושפע מסדר הקלט. בשלב הבא, הפונקציה תדפיס על המסך את כל המספרים הראשוניים שהם פולינדרומים בתחום שהזין המשתמש. לדוגמא, עבור התחום [10000,12000] התכנית תדפיס:

10301 10501 10601 11311 11411

תרגיל 3 (20 נק')

כתבו פונקציה בשם **RemoveMinDigit** המקבלת מספר שלם ארוך וחיובי, הפונקציה תחזיר את המספר המתקבל אחרי ההורדה של כל המופעים של הספרה המינימלית במספר.

למשל :

עבור המספר 3527267 הפונקציה תחזיר 35767

עבור המספר 2222 הפונקציה תחזיר 0

תרגיל 4 (20 נק')

כתבו פונקציה בשם **Game** המבצעת משחק זיכרון, לא מקבלת פרמטרים ולא מחזירה ערך. בתחילת המשחק הפונקציה תשאל את המשתמש האם ברצונו להתחיל משחקון חדש. במידה והמשתמש בחר להתחיל משחקון חדש הוא יתבקש להכניס תחום ניחוש הבנוי משני מספרים חיוביים שלמים, כאשר התחום שהתכנית תיצור אינו מושפע מסדר הקלט. הפונקציה תגדיל מספר אקראי, ולאחר-מכן הפונקציה תשאל את המשתמש לכמות הניסיונות שהוא מוכן לנסות לנחש את המספר (בין 2-10 ניחושים). בשלב הניחושים, במידה והמשתמש מזין מספר שאינו בתחום (בין שני המספרים שהוא הזין), הפונקציה תודיע בהודעה מתאימה אך תנכה את ניסיון זה ממספר הניחושים שהוקצו לו (ויציג אותם). במידה והמשתמש הזין מספר בתחום אך לא מצא את המספר הנדרש, הפונקציה תודיע לו האם המספר שלו נמוך/גבוה מהמספר הנדרש ותנכה ניסיון אחד מכמות הניסיונות שהוקצו לו (ויציג אותם). המשחקון ייעצר במידה והמשתמש ניחש את המספר הנדרש או שמספר הניסיונות שלו נגמר, בהתאם לכך תינתן למשתמש הודעה מתאימה (בהתאמה). בתום כל משחקון המשתמש ישאל האם ברצונו לשחק שוב.

דוגמא חלקית להרצת המשחק:

Enter a number : 40

Enter a number : 10

Enter the number of guesses you would like to make [2-10] : 3

לדוגמא המחשביגדיל את המספר 25.

Enter your guess (you have 3 guesses left) : 50

Invalid guess – not in the required range [10-40]

Enter your guess (you have 2 guesses left) : 20

Enter a higher number (you have 1 guesses left) : 25

Congratulations – you have won !

הסבר לגבי הגרלת המספר - השימוש בפונקציית rand :

פונקציית rand הנמצאת בספרייה `<stdlib.h>`, ולכן כדי להשתמש בה חובה לעשות `include` לספרייה הנ"ל, וכן לספרייה `<time.h>`

בנוסף, בשביל להשתמש בפונקציה הנ"ל אנו חייבים לרשום באחת השורות הראשונות בפונקציה הראשית את הפקודה:

```
srand(time(NULL)); // Initialization of random
```

כעת כאשר נרצה לקבל במיקום כלשהוא בקוד מספר אקראי נשתמש בפונקציה בצורה הנ"ל

```
int value = (rand() % (MAX_VALUE - MIN_VALUE + 1)) + MIN_VALUE;
```

כאשר MAX_VALUE הוא הערך המקסימלי שאנו חוצים לקבל (למשל 100), ו-MIN_VALUE הוא הערך המינימלי שאנו חוצים לקבל (למשל 20). אז המספר האקראי המוגדל יהיה בין (כולל) 20 ל-100 (כולל).

תרגיל 5 (20 נק')

כתבו פונקציה בשם **PrintPattern** המקבלת מספר שלם חיובי (במידה והוא שלילי להפוך אותו לחיובי) ותו . הפונקציה מדפיסה תבנית בצורת הבאה.

למשל,
עבור מספר 6 ותו '*' , התבנית תראה כך:

```
*****
*****
*****
*****
*
*
*
*****
*****
*****
*****
```