

מבוא למדעי המחשב – אביב 2023 – מטלה 3

נושאים: רקורסיה, מערך חד ממדי.

משקל מציון התרגיל: 10%

תאריך הגשה: 23:50 08.05.2023

הגשה באיחור: ניתן להגיש באיחור של יום (בהורדת 10 נקודות – ציון מקסימלי 90), יומיים (בהורדת 20 נקודות – ציון מקסימלי 80) או שלושה ימים (בהורדת 30 נקודות – ציון מקסימלי 70). לאחר מכן לא תתאפשר ההגשה (מלבד לאיחורים מוצדקים לפי תקנון האוניברסיטה).

הנחיות כלליות

שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי** בפורום הייעודי למטלה הנמצא במודל.

בקשות להארכה מסיבות מוצדקות (מילואים, לידה, אשפוז וכו') יש לשלוח למייל tom.ben-dor@biu.ac.il בצירוף: שם מלא, שם משתמש במערכת ההגשה, מספר תעודת זהות ומסמכים רלוונטיים לפי הצורך.

יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות. אין להוסיף או להשמיט רווחים או תווים אחרים, ואין להחליף אותיות גדולות בקטנות או להיפך. חוסר הקפדה על פרטים אלו עלול לגרור הורדה משמעותית ביותר בציון התרגיל עד כדי 0. ראו עצמכם הוזרתם!

שימו לב שאתם עוקבים במדויק אחרי ההנחיות במסמך ה-Style Guide המפורסם באתר הקורס.

עליכם לכתוב קוד על פי הוראות התרגיל ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על שרתי המחלקה (planet) **ללא שגיאות וללא אזהרות**. תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו הורדת נקודות משמעותית בציון שלו.

להזכיר העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. העתקות נבדקות על ידי מערכת ההגשה האוטומטית, ותרגיל שהועתק יגרור בין השאר ציון 0 ופגיעה בציוני התרגול הסופיים **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.

בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה, ולהגיש אותם באמצעות מערכת ההגשה.

בפיתוח הקוד אין להיעזר בכלים מונחי למידת מכונה מכל סוג, על כל קטע קוד להיכתב על ידי המגיש/ה בלבד ובאופן עצמאי.

דוגמת הרצה של התרגיל נמצאת בסוף המסמך.

תפריט

בתחילת התוכנית יודפס למסך התפריט הבא:

```
Choose an option:  
0. Exit  
1. Evaluator  
2. Binary strings  
3. Exciting number  
4. Cool number  
5. Count paths
```

לאחר מכן, על פי בחירת המשתמש, תתבצע המשימה המבוקשת.
בסיום ביצוע משימה, יש להדפיס מחדש את התפריט ולפעול על פי בחירת המשתמש.
במידה והמשתמש מקיש 0, יש לסיים את התוכנית.
במידה והמשתמש מקיש אופציה שאינה מופיעה בתפריט, יש להדפיס **Invalid option ולהדפיס מחדש את התפריט.**
ניתן להניח שהמשתמש יכניס תו בודד (יכול להיות כל תו) ולאחר מכן ילחץ Enter.

שימו לב: לאחר הדפסת התפריט יש לרדת שורה.

דוגמאות הרצה לתרגיל עם קלט ופלט מצופים מופיעים בסוף המסמך וכן מעודכנים במערכת הגשת התרגילים.

משימה 1 - חישוב ערך של ביטוי מקונן

לאחר שהשתמש בחר במשימה 1 - הוא יתבקש להכניס ביטוי באופן הבא:

Enter an expression:
 $(1+(2*3)+5*8+1)$

ביטוי הוא ביטוי מתמטי פשוט הבנוי ממספרים, אופרטורים (+, -, *, /) וסוגריים. הביטוי מקונן, כלומר הוא מורכב מתתי ביטויים שבתורם גם מורכבים מתתי ביטויים, וכך הלאה.

התוכנית תחשב את ערך הביטוי באופן הבא:
אין חשיבות לקדימות אופרטורים, חישובים מתבצעים משמאל לימין, עם חשיבות לסוגריים.
כלומר עבור הקלט הנ"ל, ערך הביטוי יחושב כך:

$$\left(\left((1 + (2 \cdot 3)) + 5 \right) \cdot 8 \right) + 1 = 97$$

ויודפס:

The value is 97

במידה ומסיבה כלשהי הביטוי אינו חוקי או שהוא מכיל חלוקה באפס, יש להדפיס:

Invalid expression

ולחזור לתפריט הראשי.

נקודות חשובות:

- הפתרון חייב להיות רקורסיבי. זה לא אומר שלולאות אסורות, אבל מבנה הפתרון צריך להיות רקורסיבי. הבעיה מנוסחת כך שפתרון רקורסיבי הוא הפתרון הטבעי והפשוט ביותר ליישום. רמז: איך מחשבים את ערך הביטוי $(4 + 6) \cdot (1 + 2)$? מחשבים את ערכי תתי הביטויים ברקורסיה (מה שבתוך הסוגריים) ומכפילים את התוצאות. אין להשתמש במערכים.
- המספרים המופיעים בביטוי חוקי הם בטווח $0 - 9$ כולל, כלומר ספרה אחת. זה נועד להקל על הפתרון.
- ביטוי חוקי תמיד מתחיל בסוגר פותח (וגמר בסוגר סוגר).
- אין מגבלה על אורך הביטוי, מספר תתי הביטויים בו או אופיים.
- בביטוי חוקי יכולים להופיע רווחים בהתחלה, באמצע או בסוף ויש להתעלם מהם.
- ערך כל ביטוי חוקי יהיה בטווח התקין של מספר int, אך לא בהכרח חיובי.

משימה 2 – מציאת כל הרצפים הבינאריים באורך n ללא אחדות רצופים

לאחר שהמשתמש בחר במשימה 2 – הוא יתבקש להכניס מספר טבעי באופן הבא:

Enter length:

3

התוכנית תדפיס את כל הרצפים הבינאריים (מחרוזות המורכבות מאפסים ואחדות) באורך 3 בדיוק אשר לא מכילות אחדות רצופים.
לדוגמה, עבור הקלט 3, יודפס:

000
001
010
100
101

נקודות חשובות:

- הפתרון חייב להיות רקורסיבי, במשימה זו לולאות אסורות. ניתן לכתוב "פונקציות עזר" בתנאי שגם הן רקורסיביות או ללא לולאות. אין להשתמש במערכים.
- האורך שמוכנס על ידי המשתמש מובטח להיות מספר שלם בגודל `int`, אך לא בהכרח חיובי. ניתן להניח שהתוכנית תתמוך באורך 32 – 1 כולל בלבד (חשבו איך ההנחה הזו עוזרת לכם בכתיבת התוכנית) ויש לאכוף את הטווח הזה. במידה והקלט לא בטווח הנ"ל יש להדפיס:

Invalid length

- עם ירידת שורה, ולחזור לתפריט הראשי.
- הרצפים מודפסים בסדר עולה, לפי הגודל של המספר אותו הן מייצגות, בשיטת `unsigned`.

משימה 3 – בדיקת מספר מרגש

לאחר שהשתמש בחר במשימה 3 – הוא יתבקש להכניס מספר טבעי באופן הבא:

Enter a natural number:

153

התוכנית תבדוק האם המספר הוא מספר "מרגש" או לא.
מספר מרגש מוגדר להיות מספר אשר סכום ספרותיו, כאשר כל אחת בחזקת מספר ספרותיו, שווה לו.

לדוגמה:

$$1^3 + 5^3 + 3^3 = 153$$

ולכן 153 הוא מספר מרגש, והפלט יהיה:

The number is exciting

במידה והמספר שהוכנס אינו מרגש, הפלט יהיה:

The number is not exciting

נקודות חשובות:

- הפתרון חייב להיות רקורסיבי, במשימה זו לולאות אסורות. ניתן לכתוב "פונקציות עזר" בתנאי שגם הן רקורסיביות או ללא לולאות. אין להשתמש במערכים.
- מובטח שיוכנס מספר שבטוחו משתנה `unsigned int`.

משימה 4 – בדיקת מספר מגניב

לאחר שהשתמש בחר במשימה 4 – הוא יתבקש להכניס מספר טבעי באופן הבא:

Enter a natural number:

14

התוכנית תבדוק האם המספר הוא מספר "מגניב" או לא.

כדי להבין מהו מספר מגניב, נסתכל על המספר המגניב 14:
אם ניצור סדרה שבה האיברים הראשונים הם ספרות המספר:

1,4

המספר 14 מורכב משתי ספרות, לכן האיבר הבא בסדרה יהיה הסכום של שני האיברים הקודמים לו:

1,4,5,9,14

מכיוון שהמספר 14 מופיע בסדרה, הוא מספר מגניב! הפלט יהיה:

The number is cool

לעומת זאת המספר 123 הוא לא מגניב, נסתכל על הסדרה שלו, בה סוכמים כל פעם את שלושת האיברים האחרונים:

1,2,3,6,11,20,37,68,125 ...

וברור ש-123 לעולם לא יופיע בסדרה, והפלט יהיה:

The number is not cool

נקודות חשובות:

- פתרון משימה זו לא חייב להיות רקורסיבי.
- מובטח שיוכנס מספר בטווח משתנה `unsigned int`.
- בפתרון שאלה זו תצטרכו להשתמש במערך. להזכירכם, גודל המערך צריך להיקבע בזמן הקומפילציה ולא כתלות משתנה בקוד. כלומר נגדיר בתחילת התוכנית `#define ARRAY_SIZE ?` כאשר במקום סימן שאלה יופיע מספר כלשהו, ונשתמש ב `[ARRAY_SIZE]` כשנרצה להגדיר את המערך (אם נצטרך פחות תאים מ-`ARRAY_SIZE`, נתייחס למערך כאל קצר יותר ממה שהוא במציאות). עליכם לחשוב לבד מה הערך שצריך להופיע במקום סימן השאלה. רמז: יש ערך אחד שהוא הערך הטוב ביותר, ערך זה ייבדק בבדיקה הידנית (כדי שלא תיצרו לדוגמה מערך בגודל 100000 כדי להיות בטוחים שהוא יספיק תמיד).

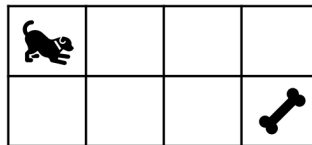
משימה 5 – ספירת מסלולים

לאחר שהשתמש בחר במשימה 5 – הוא יתבקש להכניס שני מספרים באופן הבא:

Enter width and height:

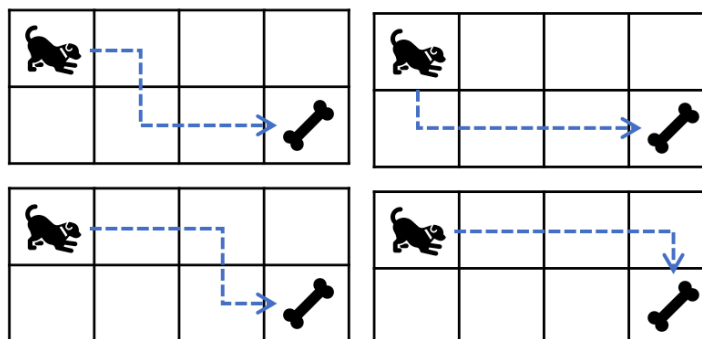
4 2

כעת נגדיר את הבעיה עבור הפרמטרים שהוכנסו:



כלבלב נמצא בפינה השמאלית העליונה של טבלה בגודל 4×2 , הוא מעוניין להגיע לעצם בפינה הימנית התחתונה, כשהוא מסוגל לנוע ימינה או למטה בלבד. עליכם לחשב את מספר המסלולים בהם יכול הכלבלב לנוע.

במקרה זה, ניתן לראות בקלות שקיימים ארבעה מסלולים:



עליכם לקחת בחשבון גם אילוץ נוסף: הכלבלב לא יכול לצעוד במסלולים אשר מכילים צעידה באותו כיוון במשך שלושה צעדים ברצף. כלומר מתוך 4 המסלולים הנ"ל, שני המסלולים הימניים נפסלים. הפלט צריך להיות:

The number of paths is 2

נקודות חשובות:

- הפתרון חייב להיות רקורסיבי, במשימה זו לולאות אסורות. ניתן לכתוב "פונקציות עזר" בתנאי שגם הן רקורסיביות או ללא לולאות. אין להשתמש במערכים.
- מובטח שיוכנסו מספרים בטווח של `unsigned int`.
- עבור טבלה בגודל $0 \times n$ קיימים 0 מסלולים, עבור טבלה בגודל $m \times 1$ קיים מסלול יחיד.

הוראות הגשה

יש להגיש את התוכנית שכתבתם בקובץ בודד בשם ex_3.c לקבוצה מספר 14.

דוגמת הרצה

כאמור, דוגמה זו היא **חלקית**, עליכם לחשוב על כלל מקרי הקצה האפשריים בתוכנית ולבדוק אותם.

```
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
1
Enter an expression:
(1+(2*3)+5*8+1)
The value is 97
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
1
Enter an expression:
((5)+(1-1))
The value is 5
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
2
Enter length:
3
000
001
010
100
101
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
3
Enter a natural number:
153
The number is exciting
```



```
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
3
Enter a natural number:
13
The number is not exciting
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
4
Enter a natural number:
14
The number is cool
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
4
Enter a natural number:
123
The number is not cool
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
5
Enter width and height:
4 2
The number of paths is 2
Choose an option:
0. Exit
1. Evaluator
2. Binary strings
3. Exciting number
4. Cool number
5. Count paths
0
```