

פרק 6 – נכונות אלגוריתמים

לאורך ההיסטוריה הקצרה של מדעי המחשב יש אינספור דוגמאות של תוכניות שגויות, ולא מעט מהן הסתיימו בכי רע. אחת הדוגמאות היא סיפורה של ספינת חלל אמריקנית מסדרת מרינר שנשלחה אל כוכב הלכת ונוס. הספינה אבדה כתוצאה משגיאה בתוכנית המחשב שהופקדה על בקרת הטיסה, ומיליוני דולרים ירדו לטמיון.

חלק לא מבוטל מן המחקר במדעי המחשב מוקדש לתחום הנקרא "הוכחת נכונות של תוכניות". מטרתה של הוכחת נכונות של תוכנית היא אימות מלא לטענה שהתוכנית מציגה עבור כל קלט את הפלט הדרוש.

חוקרים פיתחו ומפתחים שיטות שונות להוכחת נכונות של תוכניות. שיטות אלה הן מתמטיות באופיין ונשענות על תיאוריות מתמטיות מורכבות למדי. חומר הלימוד של "יסודות מדעי המחשב" איננו דן בהוכחת נכונות של תוכניות, כיוון שהדבר מחייב ידע מתמטי שאיננו נרכש בבית-ספר תיכון. אך ההתייחסות לנכונות של תוכניות חשובה לאורך חומר הלימוד כולו, כבר עם פיתוח תוכניות ראשונות. לכן, אנו מציגים נושא זה כבר עתה, בצורה פשוטה ואינטואיטיבית.

אלגוריתם לפתרון בעיה אלגוריתמית נתונה הוא **נכון** אם ביצעו מביא להצגת הפלט הדרוש עבור כל קלט חוקי (כלומר כל קלט המתאים להגדרת הבעיה).

מספר האפשרויות השונות לקלטים חוקיים הוא בדרך כלל רב, ולעתים רבות אפילו אינסופי, ולכן לא מתקבל על הדעת להיווכח בנכונותו של אלגוריתם על ידי בדיקת הפלט עבור כל קלט אפשרי. בפרקי הלימוד של "יסודות מדעי המחשב" אנו בודקים את נכונותו של אלגוריתם על ידי בדיקת הפלט עבור דוגמאות שונות של קלט.

בפרק 4 הצגנו בדיקת נכונות עבור **דוגמאות קלט מגוונות**; כלומר דוגמאות קלט אשר מאפייניהן מבטאים את מגוון הקלטים האפשריים. עתה נחדד בנקודה של בדיקת נכונות, ונשתמש בדרך כלל במושג **דוגמאות קלט מייצגות**.

דוגמאות קלט מייצגות הן דוגמאות קלט אשר כל אחת מהן מייצגת קבוצת קלטים. **בחירה ממצה** של דוגמאות קלט מייצגות היא בחירה המבטאת חלוקה ממצה של הקלטים האפשריים לקבוצות מייצגות.

בפתרון הבעיה הבאה נמחיש בחירה ממצה של דוגמאות קלט מייצגות, ונראה שימוש בדוגמאות הקלט המייצגות כדי לזהות שגיאה בתוכנית ולתקנה.

הצ'יה 1

מטרת הבעיה ופתרונה: הדגמה של חלוקת הקלטים לקבוצות אשר מאפייניהן שונים, בחירת דוגמת קלט מייצגת לכל קבוצה, ושימוש בדוגמאות הקלט המייצגות כדי לאמת את נכונותה של תוכנית וכדי לתקנה במידת הצורך.

נתונה התוכנית הבאה:

/*

קלט: שתי אותיות אנגליות גדולות
פלט: הודעה אם האותיות עוקבות זו לזו

```

*/
using System;
public class Letters
{
    public static void Main()
    {
        char letter1, letter2;
        Console.WriteLine("Enter a capital letter: ");
        letter1 = char.Parse(Console.ReadLine());
        Console.WriteLine("Enter another capital letter: ");
        letter2 = char.Parse(Console.ReadLine());
        if (letter2 == letter1 + 1)
            Console.WriteLine("The letters are consecutive");
        else
            Console.WriteLine("The letters are not consecutive");
    } // Main
} // class Letters

```

ביצוע התוכנית יביא להצגת הודעת פלט נכונה רק עבור חלק מן הקלטים האפשריים.

חלקו את הקלטים האפשריים לקבוצות, בחרו דוגמת קלט מייצגת לכל קבוצה, ותארו עבור אילו דוגמאות תוצג הודעה נכונה ועבור אילו דוגמאות תוצג הודעה שגויה. אחר-כך, תקנו את התוכנית כך שעבור כל קלט תוצג הודעה נכונה.

? מכותרת התוכנית וממשפטי הפלט ניתן לראות שמטרת התוכנית היא לבדוק אם שתי האותיות הנתונות כקלט הן אותיות אנגליות עוקבות. מהי חלוקה מתאימה של הקלטים האפשריים לקבוצות?

מתאים לחלק את הקלטים האפשריים לשתי הקבוצות הבאות:

1. קלטים שהאותיות הנתונות בהם הן אותיות אנגליות עוקבות.
2. קלטים שהאותיות הנתונות בהם אינן אותיות אנגליות עוקבות.

? החלוקה המתוארת אכן מבטאת אבחנה בין קלטים אשר יש להם מאפיינים שונים. האם חלוקה זו מספיקה? האם לא קיימים קלטים בעלי מאפיינים שונים בקבוצות אלו?

למשל, הקלט B C שייך לקבוצה הראשונה, שהאותיות בה הן אנגליות עוקבות. גם הקלט C B שייך לקבוצה הראשונה, אך לשני קלטים אלה מאפיינים שונים. בקלט B C האות השנייה עוקבת לאות הראשונה, ואילו בקלט C B האות הראשונה עוקבת לאות השנייה.

שימו ♥: ניתן גם לחלק את קבוצת הקלטים השנייה באופן דומה, אך חלוקה זו איננה נחוצה: כיוון שהאותיות אינן עוקבות, הרי סדר הופעתן בקלט איננו משנה.

אם כך, נקבל את החלוקה הבאה של הקלטים האפשריים לשלוש קבוצות:

- א. האות השנייה עוקבת לאות הראשונה.
- ב. האות הראשונה עוקבת לאות השנייה.
2. שתי האותיות אינן עוקבות.

? חילקנו את הקלטים האפשריים לחלוקה ממצה לקבוצות. מה תהיה בחירה ממצה של דוגמאות קלט מייצגות עבור הבעיה הנתונה?

בחירה ממצה של דוגמאות קלט מייצגות תהיה בחירת דוגמת קלט מכל קבוצה. למשל, קבוצת דוגמאות הקלט הבאות מבטאת בחירה ממצה: X Y (מייצגת את קבוצת הקלטים a); E D ; (מייצגת את קבוצת הקלטים b); D O ;

? עבור אלו מדוגמאות הקלט המתוארות תוצג הודעת פלט נכונה?

עבור דוגמת הקלט הראשונה (X Y) והשלישית (D O) תוצג הודעת פלט נכונה. אך עבור דוגמת הקלט השנייה (E D) תוצג הודעת פלט שגויה! עבור דוגמה זו תוצג ההודעה:

```
The letters are not consecutive
```

הודעה שגויה תוצג, בעצם, עבור כל קלט מהקבוצה b1, המיוצגת בדוגמת הקלט השנייה.

? כיצד ניתן לתקן את התוכנית, כך שתציג פלט דרוש עבור כל קלט חוקי? כלומר, כך שגם עבור קלטים מן הקבוצה השנייה תוצג הודעה מתאימה?

יש להרחיב את הביטוי הבוליאני שבמשפט ה-if. הביטוי הבוליאני שבתוכנית הנתונה כולל רק את האפשרות שהאות השנייה בקלט היא אות עוקבת לאות הראשונה בקלט. יש להרחיב ביטוי זה כך שיכלול גם את האפשרות שהאות הראשונה בקלט היא אות עוקבת לאות השנייה.

לכן, הביטוי הבוליאני שבתוכנית המתוקנת יהיה:

```
(letter2 == letter1 + 1) || (letter1 == letter2 + 1)
```

התוכנית המתוקנת תהיה:

```
/*
קלט: שתי אותיות אנגליות גדולות
פלט: הודעה אם האותיות עוקבות
*/
using System;
public class Letters
{
    public static void Main()
    {
        char letter1, letter2;
        // קלט
        Console.WriteLine("Enter a capital letter: ");
        letter1 = char.Parse(Console.ReadLine());
        Console.WriteLine("Enter another capital letter: ");
        letter2 = char.Parse(Console.ReadLine());
        if ( (letter2 == letter1 + 1) || (letter1 == letter2 + 1) )
            Console.WriteLine("The letters are consecutive");
        else
            Console.WriteLine("The letters are not consecutive");
    } // Main
} // class Letters
```

סוף פתרון הציה 1

ניתן ללמוד לקח מן הפתרון לבעיה 1:

השגיאה שהופיעה בתוכנית הנתונה נובעת מכך שמפתח התוכנית לא ביצע ניתוח מלא של כל אפשרויות הקלט השונות. מפתח התוכנית לא הבחין בכך שהמשמעות של אותיות אנגליות עוקבות, איננה רק האפשרות ש"האות השנייה עוקבת לאות הראשונה", אלא גם האפשרות ש"האות הראשונה עוקבת לאות השנייה". זיהוי השגיאה ותיקונה התאפשר הודות לחלוקה ממצה של הקלטים האפשריים לקבוצות, והיא הביאה לבחירה ממצה של דוגמאות קלט מייצגות.

שימו ♥: הבדיקה בעזרת דוגמאות קלט מייצגות אינה **מוכיחה** נכונות, אלא מסייעת באיתור שגיאות, ומקטינה מאוד את ההסתברות לטעות, אך תמיד ייתכן (בייחוד בתוכניות גדולות מאוד) שנפספס תת-מקרה מסוים, משום שאין בידינו מתכון לקביעת חלוקה ממצה לקבוצות.

כיום בכל חברת תוכנה יש צוות בודקים אשר כל תפקידו הוא לוודא כי התוכנה עובדת כשורה עבור כל קלט אפשרי. חשיבות בדיקות אלה גבוהה ונועדה על מנת להימנע מהפסדים ולעתים גם מנזקים משמעותיים הרבה יותר, כמו פגיעה בחיי אדם (למשל בתוכנות רפואיות).

במרבית חברות התוכנה מערך הבדיקות מבוסס על סימולציה, כלומר על בחירה מתוככמת של דוגמאות קלט מייצגות ובדיקת התוכניות עליהן, בחירה שיכולה להיות מורכבת מאוד ומסובכת מאוד בתוכניות גדולות ומורכבות. עם זאת, יש גם חברות (בעיקר חברות לתכנון ולפיתוח חומרה) המשתמשות בכלי ה**וכחה** (הנשענים על תורות מתמטיות). הסיבה שממעטים להשתמש בכלי ההוכחה היא שלכלים כאלה הקיימים כיום קשה להתמודד בצורה נוחה עם תוכניות גדולות. הסיבה שמשתמשים בהם יותר בתעשיית החומרה היא שהרבה יותר יקר לייצר מחדש רכיב חומרה אם מתגלית בו טעות אחרי שלב הייצור, מאשר לתקן ולהפיץ גרסה חדשה של תוכנה שהתגלתה בה טעות.

שאלה 6.1

נתון הלוח הבא, ובכל משבצת בו מופיע מספר שלם:

1	2
3	4

קטע התוכנית הבא, אשר הקלט שלו הוא שניים מן המספרים המופיעים בלוח, יציג כפלט הודעה.

```
Console.WriteLine("Enter first number: ");
num1 = int.Parse(Console.ReadLine());
Console.WriteLine("Enter second number: ");
num2 = int.Parse(Console.ReadLine());
if (num1 == num2 + 2)
    Console.WriteLine("The numbers are in the same column");
else
    Console.WriteLine("The numbers are not in the same column");
```

- בחרו שתי דוגמאות קלט, אשר עבור כל אחת מהן תוצג הודעה נכונה אחרת. תארו את המאפיינים של שתי קבוצות הקלטים שהדוגמאות שבחרתם שייכות אליהן.
- בחרו דוגמת קלט שעבורה תוצג הודעה שגויה, ותארו את המאפיין של קבוצת הקלטים שעבורם תוצג הודעה שגויה.
- שנו את הביטוי הבוליאני שבקטע התוכנית לביטוי בוליאני אחר, שעבורו תוצג תמיד הודעה מתאימה. נסחו את הביטוי החדש כביטוי פשוט.

שאלה 6.2

הקלט לקטע התוכנית הבא הוא מספר שלם כלשהו. מטרת קטע התוכנית הבא היא להציג כפלט ערך שסימנו כסימן המספר שנקלט וגודלו הוא ריבוע המספר שנקלט. למשל עבור הקלט 10 הפלט הנדרש הוא 100, ועבור הקלט 10- הפלט הנדרש הוא -100.

שימו ♥: בקטע התוכנית נעשה שימוש בפעולת החזקה Pow של המחלקה המתמטית Math. הפעולות המתמטיות הוצגו בפרק 4.

```
num = int.Parse(Console.ReadLine());
s = Math.Pow(num, 2);
Console.WriteLine(s);
```

קטע התוכנית שגוי.

- תנו דוגמת קלט מייצגת שעבורה יתקבל הפלט הדרוש.
- תנו דוגמת קלט מייצגת שעבורה לא יתקבל הפלט הדרוש.
- תארו את קבוצת הקלטים שעבורם לא יתקבל הפלט הדרוש.
- תקנו את קטע התוכנית, כך שעבור כל קלט יתקבל הפלט הדרוש.

שאלה 6.3

נתון קטע התוכנית הבא אשר הקלט שלו הוא ארבע אותיות אנגליות גדולות.

```
letter1 = char.Parse(Console.ReadLine());
letter2 = char.Parse(Console.ReadLine());
letter3 = char.Parse(Console.ReadLine());
letter4 = char.Parse(Console.ReadLine());
if ((letter1 == letter2) && (letter3 == letter4))
    Console.WriteLine("All letters are equal");
else
    Console.WriteLine("Not all letters are equal");
```

- תנו שתי דוגמאות קלט שונות אשר עבור כל אחת מהן תתקבל הודעה שגויה.
- תארו את קבוצת הקלטים שעבורם תתקבל הודעה שגויה.
- שנו את הביטוי הבוליאני שבקטע התוכנית כך שעבור כל קלט תתקבל הודעה נכונה.

שאלה 6.4

לפניכם תוכנית בשפת C#. התוכנית קולטת תו כקלט. מטרת התוכנית היא לבדוק האם התו מייצג מספר. אם כן, יוצג המספר הבא אחריו כפלט. אחרת תוצג הודעה כי התו אינו מספר. הראו **שלוש** דוגמאות קלט מייצגות לתוכנית זו והסבירו עבור כל דוגמה איזו קבוצה היא מייצגת. מצאו את השגיאה בתוכנית ותקנו אותה.

```
/*
התוכנית קולטת תו
אם התו מייצג מספר, התוכנית מציגה את המספר הבא אחריו
בכל מקרה אחר מוצגת ההודעה "לא מספר"
*/
using System;
public class NextNumber
{
    public static void Main ()
    {
        // הגדרת משתנים
        char character; //תו הקלט
        // קלט
```

```

Console.Write("Enter a character: ");
character = char.Parse(Console.ReadLine());
// פלט
if (character >= '0' && character <= '9')
    Console.WriteLine( (char)(character + 1) );
else
    Console.WriteLine("Not a number");
} // Main
} // class NextNumber

```

ראינו עד כה ניתוח של תוכנית נתונה שמטרתה ברורה. אך בפיתוח תוכנה קורה לא פעם שיש לשלב תוכניות אשר אינן מתועדות כראוי ומטרתן איננה ברורה. במקרים כאלו יש לזהות קודם כל את מטרת התוכנית הנתונה.

זיהוי מטרת תוכנית נתונה מתבצע על ידי כך שנבחן את פלט התוכנית עבור דוגמאות קלט שונות, ונקבע את היחס בין הקלט לפלט.

נדגים זאת בעזרת שתי השאלות הבאות.

שאלה 6.5

נתון קטע התוכנית הבא שהקלט שלו הוא מספר לא שלילי, והמשתנים בו הם מטיפוס ממשי. **שימו** ♥: בקטע התוכנית נעשה שימוש בפעולות `Math.Sqrt` ו-`Math.Floor` של המחלקה המתמטית `Math` שהוצגה בפרק 4. הפעולה `Math.Floor` מקבלת מספר ממשי ומחזירה את החלק השלם שלו (למשל, ערך הביטוי `Math.Floor(5.8)` שווה ל-5.0).

```

num = double.Parse(Console.ReadLine());
sqrtNum = Math.Sqrt(num);
fraction = sqrtNum - Math.Floor(sqrtNum);
if (fraction > 0)
    Console.WriteLine("1");
else
    Console.WriteLine("0");

```

א. תנו דוגמת קלט שעבורה יהיה הפלט 1.

ב. תנו דוגמת קלט שעבורה יהיה הפלט 0.

ג. תארו את מטרת קטע התוכנית וחלקו את הקלטים האפשריים לשתי קבוצות. תארו את הפלט עבור הקלט בכל קבוצה.

שאלה 6.6

נתון קטע התוכנית הבא, שהקלט שלו הוא ארבעה מספרים שלמים שונים. כל המשתנים בקטע התוכנית הם מטיפוס שלם.

```

num1 = int.Parse(Console.ReadLine());
num2 = int.Parse(Console.ReadLine());
num3 = int.Parse(Console.ReadLine());
num4 = int.Parse(Console.ReadLine());
if (num1 > num2)
    max1 = num1;
else
    max1 = num2;
if (num3 > num4)
    max2 = num3;

```

```

else
    max2 = num4;
if (max1 > max2)
    Console.WriteLine(max1);
else
    Console.WriteLine(max2);

```

- א. מהו הפלט עבור כל אחת מדוגמאות הקלט הבאות: 10 20 30 40 ו-50 30 40 20.
 ב. תנו שלוש דוגמאות קלט שונות של מספרים חיוביים שהפלט עבורן הוא 5.
 ג. מהי מטרת קטע התוכנית?

שאלה 6.7

לפניכם תוכנית בשפת C#:

```

/* התוכנית קולטת שלושה מספרים שלמים
_____ */
using System;
public class What
{
    public static void Main ()
    {
        // הגדרת משתנים
        int num1, num2, num3; // משתני הקלט
        int temp;           // משתנה עזר
        //קלט
        Console.Write("Enter first number: ");
        num1 = int.Parse(Console.ReadLine());
        Console.Write("Enter second number: ");
        num2 = int.Parse(Console.ReadLine());
        Console.Write("Enter third number: ");
        num3 = int.Parse(Console.ReadLine());
        if (num1 > num2)
        {
            temp = num1;
            num1 = num2;
            num2 = temp;
        }
        if (num2 > num3)
        {
            temp = num2;
            num2 = num3;
            num3 = temp;
        }
        // פלט
        Console.WriteLine("{0},{1},{2}", num1, num2, num3);
    } // Main
} // class What

```

- א. כתבו במשפט אחד מהי לדעתכם מטרת התוכנית.
 ב. הציעו חלוקה ממצה של הקלטים האפשריים לקבוצות.
 ג. בחרו דוגמאות קלט מייצגות על פי החלוקה שהצעתם בסעיף א', וציינו את הפלט עבור כל אחת מהן.
 ד. האם התוכנית משיגה את מטרתה? אם כן, הסבירו מדוע; אם לא, תקנו אותה.

סיכום

בפרק זה הרחבנו והעמקנו בנושא נכונות של אלגוריתם.

אלגוריתם נכון הוא אלגוריתם אשר ביצעו מביא להשגת המטרה עבור **כל** קלט חוקי (המתאים להגדרת הבעיה).

לא מתקבל על הדעת להיווכח בנכונות של אלגוריתם על ידי בניית טבלת מעקב עבור כל קלט אפשרי.

לכן אנו בוחרים בצורה ממצה דוגמאות מייצגות של קלט ובודקים את מהלך ביצוע האלגוריתם עבור דוגמאות קלט אלה.

דוגמאות קלט מייצגות הן דוגמאות קלט אשר כל אחת מהן מייצגת קבוצת קלטים בעלת מאפיינים שונים.

בחירה ממצה של דוגמאות קלט מייצגות היא בחירה המבטאת חלוקה ממצה של הקלטים האפשריים לקבוצות מייצגות.

לפעמים נתונים תוכניות או קטעי תוכניות אשר מטרתם אינה ברורה. **זיהוי מטרת תוכנית נתונה** נעשה לפי בחינת פלט התוכנית עבור דוגמאות קלט שונות, ולפי הכללת היחס בין הקלט לפלט.