



חוברת מבוא לתכנות

תוכן עניינים

3 אלגוריתמיקה

3 איך נותנים הוראות למחשב ?

5 מאפייני האלגוריתם

8 שבעת הכללים עליהם נקפיד בכתיבת אלגוריתם

9 תאי זיכרון

10 תאי זיכרון קבועים ומשתנים

12 ביטוי חשבוני ופעולות כלליות

16 ריצה יבשה

17 טבלת מעקב

אלגוריתמיקה

איך נותנים הוראות למחשב ?

המחשב הוא "גולם", ובשביל לבצע פעולות (אפילו הפשוטות ביותר) יש לתת לו הוראות מדויקות כיצד לבצע זאת.

חשיבות הדיוק בהוראות המחשב גבוהה מאוד. נאמר ויש לנו רובוט משוכלל בבית, אם נבקש מהרובוט המשוכלל להשליך את האשפה: "רובוט השלך את האשפה", ייתכן והרובוט ייקח את שקית האשפה וישליך אותה דרך החלון; לכן יש לומר לרובוט: "רובוט קח את שקית האשפה, צא החוצה והשלך את השקית לפח האשפה שבחוץ".

אלגוריתם (Algorithm) - אוסף סופי וחד משמעי של הוראות שביצוען, זו אחרי זו, מוביל לפתרון בעיה מוגדרת מראש.

ישנם אלגוריתמים מוכרים בהם אנו משתמשים ביומיום מבלי לדעת כי אלה אלגוריתמים:

שם האלגוריתם	הסבר
חציית כביש	תמיד מלמדים אותנו את האלגוריתם : עמוד לפני שפת הכביש הסתכל ימינה הסתכל שמאלה הסתכל שוב ימינה אם הכביש פנוי התחל לחצות את הכביש המשך להסתכל לצדדים עד הגעתך לצד השני
הוראות הפעלה לוידאו	כיצד לכוון תחנה בוידאו או לכוון את השעון ? כל זאת מפורט במדריך למשתמש של הוידאו במבנה של אלגוריתם.
מעלית	באלגוריתם המעלית נצפה שלחיצה על כפתור "קרקע" תמיד יביא אותנו לקומת קרקע, ואם אנו נמצאים בקומת קרקע לא נזוז וכדומה.

למעשה כל התהליכים הממוחשבים והמכאניים שאנו נתקלים בהם הינם אלגוריתמים.

מאפייני האלגוריתם

להלן אלגוריתם:

הכנת עוגה

התחל

קח 3 כוסות קמח
הכנס לקערה גדולה
קח 2 ביצים
הוסף לקערת הקמח
קח כוס סוכר
הוסף לקערת הקמח
ערבב היטב במשך 5 דקות
שמן תבנית
הכנס את העיסה לתבנית
חמם תנור ל-250 מעלות
הכנס את התבנית לתנור
אפה במשך 30 דקות
הוצא מהתנור

סיים

האלגוריתם מראה לנו כיצד להכין עוגה. זהו אלגוריתם פשוט, שלפיו אפילו אלה מבינינו שלא יודעים לבשל – יצליחו להבין ולהכין לפיו עוגה.

אך האם האלגוריתם מספיק מפורט כך שגם המחשב יוכל להכין עוגה בקלות?
אנחנו מחלקים את האלגוריתמים לשני סוגים עיקריים – **אלגוריתמים אנושיים ואלגוריתמים תכנותיים**.

אלגוריתם אנושי - אלגוריתם אשר על מנת להפעילו יש להשתמש בשיקול דעת ואינטואיציה.
אלגוריתם תכנותי - אלגוריתם המכיל פקודות ברורות, מוגדרות ומובנות למכונה אשר תוצאת ביצוען ידועה מראש באופן חד משמעי.

להלן מספר דוגמאות שידגישו את מאפייני האלגוריתם:

הכנת כוס תה

התחל

שים בספל שקית תה
הוסף **קצת** סוכר
הרתח **מספיק** מים
הוסף לכוס
בחש והגש

סיים

בעיה: ניתן לראות כי האלגוריתם אנושי מאחר ואין הוראות מדויקות (**קצת** סוכר, **מספיק** מים).

אלגוריתם חייב להיות **חד משמעי**

ניקיון שערי אצטדיון (1)

התחל

לך לשער מס 1
רוקן את הפח ליד השער
טאטא ברדיוס של 3 מטר מהשער
עבור במטלית יבשה על פני השער
עבור אל השער הבא הנמצא בכיוון השעון
חזור להוראה מס' 2

סיים

בעיה: כאן ניתן לראות אלגוריתם מאוד מפורט של הפעולות שיש לעשות. שימו לב להוראה השישית, המורה לחזור להוראה השניה. כך נגרום למי שיבצע את ההוראות לעבור משער לשער ולבצע את המוטל עליו בכל אחד מהשערים. בגלל ההוראה השישית האלגוריתם אינו **סופי** ולכן לעולם לא ייגמר.

ניקיון שערי אצטדיון (2)

התחל

לך לשער מס 1
רוקן את הפח ליד השער
טאטא ברדיוס של 3 מטר מהשער
עבור במטלית יבשה על פני השער
עבור אל השער הבא הנמצא בכיוון השעון
אם אינך נמצא בשער מס' 1 אזי
חזור להוראה מס' 2
אחרת
לך לקבל הוראות נוספות

סיים

בעיה: בדומה לאלגוריתם הקודם, גם כאן ישנו פירוט נרחב. שימו לב כי בניגוד לאלגוריתם הקודם בו מנקים את השערים של האצטדיון עד אין סוף, באלגוריתם זה נוסף תנאי שאם השער בו המבצע נמצא הוא שער 1, סימן שהסתיים סבב ניקוי השערים וניתן להפסיק וללכת לקבל הוראות נוספות.

ניתן לראות כי **יש סוף לאלגוריתם**, להבדיל מהאלגוריתם הקודם.

אלגוריתם חייב להיות **סופי בביצוע**.

נעילת נעל

התחל

נעל את הנעל
שרוך את השרוכים
גרוב גרב

סיים

בעיה: האלגוריתם מאוד מבלבל, האם באמת לגרוב את הגרב לאחר שנעלנו את הנעל?! ישנה חשיבות רבה לסדר ההוראות באלגוריתם, ויש לשים לב שלא לתת הוראות בסדר הלא נכון.

אלגוריתם חייב להיות בעל **רצף פעולות הגיוני**.

בדיקת חולה

התחל

פתח את הפה
הוצא לשון
אם הגרון אדום אז
חולה
אחרת
בריא

סיים

בעיה: האלגוריתם מתאר בדיקה פשוטה שלאחריה **הסקת מסקנה**. קבענו כי אם הגרון אדום אז האדם חולה ואם הוא לא אדום אז הוא בריא. אך המסקנה כללית מדי מאחר ולא תמיד כשחולים הסימפטומים הם גרון אדום.

אלגוריתם חייב להסיק מסקנות מדויקות ככל שניתן.

מסירת הודעה טלפונית

התחל

גש לטלפון הסמוך
הרם השפורפרת והצמד לאוזן
האזן לקו החיוג
הושט יד לחוגה
חייג ספרה 3
חזור על פעולה מס' 5 שש פעמים
המתן עד שתענה
אמור את ההודעה שנתבקשת למסור

סיים

בעיה: ניתן לראות באלגוריתם זה כי אמנם הוא מפורט מאוד, אך רמת הפירוט עדיין לא מספקת – ומה יקרה אם הקו יהיה תפוס או מנותק או בכלל יהיה צליל של "שיחה ממתינה"?

האלגוריתם חייב לענות על כל הדרישות ולכסות את כל האפשרויות הקיימות במהלך ביצועו.

התעמלות בוקר

התחל

הרם יד ימין לצד הגוף
הרם יד שמאל לצד הגוף
הרם שתי ידיים למעלה
הרם שתי רגליים למעלה

סיים

בעיה: כאן ניתן לראות אלגוריתם הנותן הוראות שלא ניתנות לביצוע, הרמת 2 רגליים למעלה. אלגוריתם חייב להיות **בר ביצוע למחשב**.

שבעת הכללים עליהם נקפיד בכתיבת אלגוריתם:

1. אלגוריתם **ח**ד משמעי.
2. אלגוריתם **ס**ופי בביצוע.
3. רצף פעולה **ה**גיוני.
4. אלגוריתם המסיק מסקנות **מ**דויקות ככל שניתן.
5. רמת **פ**ירוט המתאימה למחשב.
6. אלגוריתם העונה על כל ה**א**פשרויות והדרישות.
7. הוראות **ב**רות ביצוע למחשב.

נזכור את כל הכללים על פי ראשי התיבות "**חסה מפאב**".

האלגוריתם הוא למעשה רצף ההוראות אותו מבין המחשב ומבצע. תהליכים רבים בחיינו הם דמויי אלגוריתם ואלגוריתמים רבים שגורים בחיינו.

תאי זיכרון

דיברנו בפרק 2 על זיכרון המחשב בצורה כללית. בסעיף זה נדבר על השימוש שלנו, כתוכניתנים, בזיכרון המחשב.

זיכרון המחשב דומה לארון גדול ובו המון מגירות כאשר בכל מגירה ניתן לשים חפץ כלשהו.

נניח ובארון שלנו אנחנו יכולים לשים פתקים שונים שעליהם אנחנו כותבים דברים שונים. כל פתק ניתן לשים במגירה אחת. לדוגמא, החלטנו לשים פתק ועליו כתוב המספר 13 במגירה הכי שמאלית בשורה האחרונה.

כאשר נקביל את הארון למחשב נראה כי:

1. הארון כולו – זיכרון המחשב (Computer Memory).
2. מגירה בארון – תא בזיכרון המחשב (Memory Cell).
3. פתק – ערך (Value) אותו שומרים בזיכרון המחשב.

על כל מגירה (תא זיכרון) ניתן לבצע שלוש פעולות בסיסיות:

1. הכנסת פתק חדש למגירה – ניתן להכניס ערך חדש לגמרי למגירה או לשנות את הערך שקיים.
2. העתקת (שכפול) פתק ממגירה אחת למגירה שניה.
3. הצגת פתק הנמצא במגירה מסויימת.

מגירה בארון שלא שמנו בו פתק חדש יכול להכיל פתקים ישנים שעליהם ערכים שונים שמישהו אחר הניח בעבר. הפתקים הנמצאים במגירות אלו הם למעשה זבל, כי הערך המוצג עליהם אינו רלוונטי עבורנו, אלא עבור מי שהניח אותם בעבר.

כך תא זיכרון שלא הכנסנו אליו ערך חדש ואיננו יודעים מה יש בתוכו, נקרא **תא זיכרון מזובל**.

שימו לב טוב טוב לכך שתא זיכרון שלא השתמשנו בו באלגוריתם **הנוכחי** הוא תא זיכרון מזובל.

אסור להסתמך על ערכיו.

תאי זיכרון קבועים ומשתנים

מה ההבדל בין תא בזיכרון המכיל את ערך תאריך יום הולדתך לבין תא זיכרון המכיל את גילך?
 ערך הראשון קבוע ואינו משתנה, לעומת ערך השני גדל בכל שנה ואינו קבוע.
תא קבוע (Constant Cell) - תא זיכרון שערכו אינו משתנה לאורך האלגוריתם.
תא משתנה (Variable Cell) - תא זיכרון שערכו יכול להשתנות במהלך האלגוריתם.

שמות קבועים ומשתנים:

שמתי את הטלפון של הפיצריה במגירה השנייה מצד שמאל בשורה השביעית מלמעלה. במקום לתאר בכל פעם למי שרצה את הטלפון איפה הוא יושב "במגירה השניה מצד שמאל בשורה השביעית מלמעלה" החלטתי לשים מדבקה על המגירה ולרשום בה "פיצה האט". ככה כל מי שחיפש את הטלפון ידע איפה הוא שמור בדיוק.

יותר קל לנו לזהות דברים ולזכור אותם עם שמות ולא לפי תיאור שלהם. ככה גם עם אנשים – יהיה לי יותר קשה להסביר למישהו על מי אני מדבר אם אני אתאר את צורתו החיצונית במקום פשוט להגיד את שמו.

לכל תא זיכרון כמו המגירה ניתן שם. יש כמה כללים שצריך למלא כאשר נותנים לתא זיכרון שם:

1. התווים החוקיים הם ספרות (0-9), אלפבית (A-Z, a-z), קו תחתון ("_") ו NATIONALS (&, %, \$, @).
2. שם של תא זיכרון יתחיל תמיד באלפבית.
3. השם חייב להיות משמעותי כדי שנוכל להבין את תפקיד המשתנה או הקבוע על פי שמו.
4. על מנת להבדיל בין משתנה לקבוע באלגוריתם שלנו – שמות משתנים יהיו באותיות קטנות (רק האות הראשונה בכל מילה תהיה גדולה) ושמות קבועים יהיו באותיות גדולות בלבד.

להלן דוגמאות:

שם הקבוע המייצג את תאריך יום הולדתי יהיה MY_BIRTHDATE. שם הקבוע מורכב מאותיות גדולות בלבד וניתן להבין משמו את משמעותו.
 אם היינו קוראים לקבוע MBD, לא ניתן היה להבין מה משמעותו ללא הסבר מילולי.
 שם המשתנה המתאר את מספר החיילים בצבא יהיה NumOfSoldiers. שם המשתנה מורכב מאותיות גדולות בתחילת כל מילה ואותיות קטנות בשאר השם.
 לא נקרא למשתנה Num מאחר ומשמעות המשתנה אינה מספקת - אנו מבינים שהוא מתאר מספר, אבל של מה?

סוגי משתנים וקבועים:

מה ההבדל בין משנה ששומר את שם המאכל שאני הכי אוהב ומשתנה ששומר את מספר הפעמים שהייתי בחו"ל?

המשתנה ששומר את שם המאכל האהוב עלי הינו משתנה **מחרוזתי** ואילו השתנה ששומר את מספר הפעמים שהייתי בחו"ל הינו משתנה **נומרי – מספרי**.

גם קבועים יכולים להיות מספריים או מחרוזתיים.

הגדרת משתנים וקבועים:

השימוש בתאי זיכרון במהלך תוכניות ואלגוריתמים דורש דבר אחד נוסף, הגדרת תאי הזיכרון בהם נשתמש.

נניח שלרשותנו ארון גדול מאוד שחבל יהיה לנו לבזבזו לאדם אחד בלבד. נאפשר למספר אנשים לעשות בו שימוש. ישנו מנגנון מסוים המנהל את חלוקת המגירות בארון ודואג ששני אנשים לא יקבלו את אותה המגירה, שהרי במצב זה אם קיבלתי מגירה ויש בה פתק שלא אנחנו שמנו נחשוב שזוהי מגירה מזובלת ונכניס פתק אחר למגירה במקום הפתק הקיים שבו משתמש מישהו אחר.

כאשר אנו מבקשים ממנגנון חלוקת המגירות מגירה חדשה, אנו נותנים למגירה שם, כי אנחנו לא רוצים לקרוא למגירות במספרים או לתאר אותן עפ"י המיקום שלהן בארון.

פעולות הקצאת תא הזיכרון ונתינת שם לתא מתרחשות גם בזיכרון ונקראות הגדרת תא זיכרון (Memory Cell Definition). לאחר הגדרת התא ניתן לפנות אליו בשמו.

הגדרת תאי הזיכרון נעשית עוד לפני תחילת האלגוריתם, כאשר ההחלטה אם תא מסויים הוא משתנה או קבוע נקבעת כבר בשלב ההגדרה.

משתנים נגדיר בטבלה הבאה:

הגדרת משתנים		
שם משתנה	טיפוס משתנה (מספרי, מחרוזתי)	תפקיד
UserAge	מספרי	גיל המשתמש
CurrentYear	מספרי	השנה הנוכחית
UserLastName	מחרוזתי	שם המשפחה של המשתמש.

קבועים נגדיר בטבלה הבאה:

הגדרת קבועים			
שם קבוע	טיפוס	ערך	תפקיד
DATE_OF_BIRTH	מחרוזתי	14.6.1982	תאריך יום ההולדת שלי.

שימו לב שבהגדרת הקבוע כבר ניתן לו ערך מכיוון שלא ניתן לשנות את ערכו במהלך האלגוריתם.
קבוע חייב להיות מאותחל בהגדרה.

אחרי שהבנו בואו נסכם את ההבדלים בין קבוע למשתנה:

קבוע	משתנה	קריטריון
מקבל שם וערך בזמן הגדרתו	מקבל רק שם בזמן הגדרתו	שלב ההגדרה
ערכו אינו ניתן לשינוי במהלך האלגוריתם	ערכו ניתן לשינוי במהלך האלגוריתם	האם הערך ניתן לשינוי?
נפנה אל ערכו על ידי ציון שם הקבוע	נפנה אל ערכו על ידי ציון שם המשתנה	אופן הגישה לערך

ביטוי חשבוני ופעולות כלליות

כעת לאחר שהגדרנו את המשתנים והקבועים ושמנו בהם ערכים נוכל לבצע פעולות על ערכים אלו. בטבלה 3.3 מפורטים מושגי יסוד בתחום הביטויים החשבוניים:

אופרנד	מספר, קבוע מספרי או משתנה מספרי.	MY_BIRTHDATE, MyAge, 5
אופרטור	פעולות חשבון	כפל (*)
		חיבור (+)
		חילוק (/)
		חיסור (-)
		סוגריים ()
		שארית (%) *
ביטוי חשבוני	1. אופרנד 2. אופרנדים המופרדים ביניהם ע"י אופרטורים	1. MyAge, 9
		2. (MyAge+1)*4
		ביטוי אונרי
אופרטור מסוג +, -, המוצמד לאופרנד אחד	-MyAge, -19, +8	

* האופרטור % מחזיר את השארית מפעולת החילוק של שני האופרנדים. לדוגמא : $3 \% 7 = 1$, $4 \% 2 = 0$

** כדי לבצע חזקה נשתמש בעברית מבנית באופרטור **
לדוגמא $5**2 = 32$. בתוכניות שנרשום על המחשב לא נוכל להשתמש באופרטור זה.

סדר הקדימויות של האופרטורים זהה לסדר הביטויים החשבוניים במתמטיקה:

1. חישוב סוגריים.
2. חישוב חזקה.
3. חישוב כפל וחילוק.
4. חישוב חיבור וחיסור.

אם קיים אופרטור זהה, סדר החישוב משמאל לימין.
את האופרטורים החשבוניים ניתן לבצע רק על תאים מטיפוס מספרי ולא על תאים מטיפוס מחרוזתי.

פעולות בסיסיות:

עד כה למדנו מספר פעולות בסיסיות שניתן לבצע על תאי זיכרון: השמת ערך, העתקת ערך והצגת ערך קיים.

בכתיבת אלגוריתם נשתמש בסימנים ופקודות בכדי לבצע פעולות:

לביצוע השמת ערך נשתמש בסימן = (שווה), לדוגמא בכדי לשים את הערך 13 במשתנה Number נכתוב:

```
Number = 13
```

לביצוע העתקת ערך מהמשתנה Number אל המשתנה AnotherNumber נכתוב :

```
AnotherNumber = Number
```

ניתן גם לשים ביטוי חשבוני בתוך משתנה :

```
AnotherNumber = (Number * 10) + 13
```

שגיאות נפוצות בפעולת ההשמה שיש לשים לב לא לבצע:

- יש לשים לב שהביטוי החשבוני לא מבצע חלוקה ב 0 .
- פעולת השמה יכולה להכיל רק משתנים מאותו הסוג. יש לשים לב שלא מבצעים פעולת השמה עם מספרים ומחרוזות יחד.
- מצד שמאל של סימן ה = יכול להופיע רק משתנה ולא ביטוי חשבוני מורכב.
- מצד ימין של סימן ה = לא יכול להופיע משתנה מזובל.

פקודת פלט:

פקודת הפלט היא אמצעי התקשורת עם המשתמש היא מאפשרת להציג הודעות למשתמש, להציג את תוצאת האלגוריתם ועוד. לדוגמא כדי להציג ערך הקיים במשתנה number נכתוב :

```
הדפס Number
```

בעזרת פקודת הפלט ניתן להדפיס גם מחרוזות :

```
הדפס "בוקר טוב !"
```

לאחר פעולות פלט ירשמו הערכים לשטח מיוחד הנקרא "פלט", למשל לאחר הפקודה האחרונה יכיל שטח הפלט:
בוקר טוב !

ניתן להדפיס מספר ערכים באותה הפקודה לדוגמא:

הדפס "המספר הוא ": Number ,

בין משתנה למשתנה או בין המחרוזת למשתנה יהיה פסיק. כאשר מדפיסים מספר ערכים באותה הפקודה ההדפסה נעשית מימין לשמאל.
נניח ש Number שווה ל 4 , אז יודפס:

המספר הוא 4

ניתן להדפיס גם ביטויים חשבוניים לדוגמא:

הדפס Number * 4

במקרה זה יחושב הביטוי ורק אח"כ יודפס :

16

בכדי לרדת שורה בהדפסה נשתמש בפקודה "קפוץ שורה". גם היא יכולה לבוא בפני עצמה :

הדפס קפוץ שורה

או ביחד עם ערכים אחרים:

הדפס "המספר הוא ": , קפוץ שורה , Number

הפלט:

המספר הוא

4

שגיאות נפוצות בפקודת הפלט שיש לשים לב לא לבצע:

1. לא לשכוח את הפסיקים .
2. יש לשים לב לא להדפיס ערך של משתנה מזובל.

פקודת קלט:

פעולה נוספת שנרצה להשתמש בה במהלך כתיבת האלגוריתמים היא פקודת הקלט :
פקודת הקלט מאפשרת לבצע את אותו האלגוריתם בכל ריצה על נתונים שונים.

בפקודת הקלט עוצר המחשב את ביצוע האלגוריתם ומבקש מהמשתמש להכניס נתון כלשהוא לתוך המשתנה.

קלוט Number

גם את פקודת הקלט ניתן לבצע על מספר משתנים יחד כאשר גם כאן ההפרדה תעשה באמצעות פסיק:

קלוט NumberOfMissedCalls , NumberOfStudents

לפני כל פקודת קלט תופיע פקודת פלט שמתארת למשתמש מה אנחנו מצפים שהוא יקליד עכשיו. תחשבו שמישהו היה בא ומושיט את היד אליכם ללא כל בקשה קודם, לא יכולתם לדעת מה הוא רוצה ולכן לא יכולתם לתת לו את מבוקשו.

לכן נעשה זאת בצורה הבאה:

הדפס "הכנס בבקשה את מספר החניכים במחלקה"

קלוט NumberOfStudents

שגיאות נפוצות בפקודת הקלט שיש לשים לב לא לבצע:

1. לא לשכוח את הפסיקים .
2. יש לשים לב להתאים את סוג המשתנה לקלט המתקבל.
3. לא לשכוח את הבקשה לקלט (הפלט) עוד לפני הקלט.

ריצה יבשה

ביצוע האלגוריתם לפי סדר הפקודות ומעקב ורישום אחר כל תוצאות ביצוע הפקודות – נקרא ריצה יבשה, זאת משום שהיא מבוצעת באופן ידני- "על יבש".

ישנו צורך לבצע ריצה יבשה על גבי אלגוריתם כלשהו שכתבנו על מנת למצוא טעויות באלגוריתם.

שני מושגים הקשורים לריצת תוכנית הם:

1. **באג** (Bug) - כל מצב בו האלגוריתם שלנו אינו מתנהג כמו שאנו מעוניינים, כלומר במהלך ביצועו מתגלית טעות.

2. **דיבאג** (Debug) – תהליך תיקון האלגוריתם כך שלא יהיו בו באגים.

הדרך לביצוע ריצה יבשה היא מעבר על האלגוריתם תוך כדי שימוש בטבלת מעקב המחזיקה את ערכי המשתנים בכל שלבי האלגוריתם.

להלן דוגמא:

האלגוריתם קולט מהמשתמש את גילו ואת השנה הנוכחית ומחשב מהי שנת הלידה של המשתמש.

הגדרת המשתנים:

שם משתנה	טיפוס משתנה (מספרי, מחרוזתי)	תפקיד
UserAge	מספרי	גיל המשתמש
CurrentYear	מספרי	השנה הנוכחית
UserBirthYear	מספרי	שנת לידת המשתמש

אלגוריתם לחישוב שנת לידת המשתמש

התחל

הדפס "הכנס את גילך: "

קלוט UserAge

הדפס "הכנס את השנה הנוכחית: "

קלוט CurrentYear

$UserBirthYear = CurrentYear - UserAge$

הדפס "שנת הלידה שלך היא: ", UserBirthYear

סיים

לפני בדיקת האלגוריתם אנו בוחרים ערכים אותם נקלוט אל המשתנים בפקודות הקלט המופיעות באלגוריתם.

נניח כי הערכים שנבחרו הם:

גיל המשתמש – 22

השנה הנוכחית – 2002

טבלת מעקב

יש לשים לב כי אנו רושמים את הנתונים כפי שהם נראים לאחר כל פקודה.

שלב	UserAge	CurrentYear	UserBirthYear
1	מזובל	מזובל	מזובל
2	22	מזובל	מזובל
3	22	מזובל	מזובל
4	22	2002	מזובל
5	22	2002	1980
6	22	2002	1980

שימו לב: בכל שלב שבו משתנה עוד לא אותחל נרשום "מזובל" ולא נשאיר ריק.

הפלט יראה כך:

הכנס את גילך :

הכנס את השנה הנוכחית :

שנת הלידה שלך היא 1980

לאחר הרצת האלגוריתם אנו בודקים אם התוצאה בפלט היא התוצאה אליה ציפינו.

ניתן לראות כי אלגוריתם זה אינו מחשב במדויק את שנת הלידה מאחר ואינו מתייחס גם לחודשים ולימים בשנה (בדוק בעצמך את האלגוריתם).