

חלק יבש

לפניכם מספר שאלות, חלקן על דברים שראינו בכתה וחלקן לא. כתבו את הפתרונות אליהן בקובץ **dry.pdf**.

שאלה 1

המתרגלים בקורס שפות תכנות רצו לכתוב שאלות למבחן בקורס, אך מרוב שהשקיעו בשאלות שיעורי הבית נגמרו להם הרעיונות הטובים לשאלות. הציעו 2 שאלות בִּרְרָה למבחן (שאלות אמריקאית) על אודות חומר הלימוד שהוצג בהרצאות או בשקפים. על כל שאלת בררה להציע בדיוק 4 תשובות אפשריות, שמתוכן על הנבחן לברור תשובה אחת, הנכונה ביותר. לכל שאלה ציינו מה התשובה הנכונה ונמקו. טיפים - ראו הנחיות לכתיבת שאלות בררה מוצלחות (חלק ג' במסמך [הזה](#)).

שאלה 2

המתרגל האחראי בקורס שפות תכנות קצת שכח איך לעבוד עם מספרים בפרולוג (מבוסס על סיפור אמיתי). בשאלה זו נעזור לו לחדד את ההבדלים בין אופרטורים שונים בשפה עבור מספרים.

ענו על כל אחד מהסעיפים והסבירו בקצרה:

1. בתרגול ראינו את האופרטורים $=$, is ו- $==$.

a. איזה אופרטור מבין אלו אינו סימטרי (כלומר, לא ניתן להחליף את סדר הארגומנטים ולקבל תוצאה זהה)?

b. תנו דוגמה לשימוש ב- $==$ בה לא ניתן להשתמש באופרטור $=$ במקום.

c. תנו דוגמה לשימוש ב- $=$ בה לא ניתן להשתמש באופרטור $==$ במקום.

2. בתרגול ראינו את האופרטור $\#$ מהספריה CLPFD. עבור אילו מקרים אופרטור זה הוא הכללה של שלושת האופרטורים מהסעיף הקודם? (כלומר, תחת אילו תנאים ניתן להחליף כל אחד מהקודמים באופרטור זה)

שאלה 3

בתרגול למדנו על שפת התכנות TypeScript כדוגמה לשפה עם טיפוסיות הדרגתית (Gradual Typing).

1. ראינו שב-TypeScript ניתן להגדיר טיפוס Record באופן הבא:

```
type Record = { x: number, y: number };
```

האם תאימות בין טיפוסים כאלו הוא Nominal או Structural? מה לגבי תאימות בין טיפוסים המוגדרים על ידי מחלקות? הסבירו והביאו דוגמאות קוד התומכות בטענה שלכם.

2. בתרגול ראינו שמערכת הטיפוסים של השפה אינה בטוחה (safe), מכיוון שניתן להשתמש ב-any כדי לבטל בדיקות טיפוסים והטיפוסים אינם נבדקים בזמן ריצה. בנוסף לכך, מערכת הטיפוסים של השפה אינה בטוחה גם ללא שימוש ב-any. הביאו דוגמת קוד בה **לכל** הערכים יש טיפוס **סטטי קונקרטי** מוגדר, אך בזמן ריצה עדיין תתרחש שגיאת טיפוס.

הערות:

- תוכלו להשתמש ב-[TypeScript Playground](#) כדי לנסות את הדוגמאות שלכם.
- אין להשתמש ב-type assertions (המקביל ב-TypeScript להמרות טיפוסים).

רמזים לסעיף 2:

a. בהינתן טיפוס B היורש מ-A, הקומפיילר מתייחס למערך של הראשון כתת-טיפוס של מערך של השני, כלומר קטע הקוד הבא חוקי:

```
let bArr: B[] = [ new B(), new B() ];  
let aArr: A[] = bArr;
```

b. כאשר B תת-טיפוס של A, ניתן גם לבצע השמות של B למשתנים מסוג A וגם להעביר ערך מסוג B לפונקציה המצפה לקבל ארגומנט מסוג A.

c. עבור טיפוס נוסף C היורש מ-A (ולא מ-B), נסו לגרום לכך שיכנס איבר מסוג C למערך של איברים מסוג B.

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס - [Homework5](#).
- פתרון לכל אחת מהשאלות יש לכתוב בקובץ ששמו `hw5_q1.pl` עבור שאלה `i`.
- לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם לשאלה 1 תוכלו לבדוק בעזרת הפקודות הבאות:

```
swipl -q hw5_q1.pl < q1.in > q1.out  
diff <(sed '1,/===TEST START===/{N;d}' q1.out) q1.expected
```

- יש לבדוק ולהריץ את פתרונותיכם לתרגיל זה בדוקר **twyair/safot-hw:5**
- [מדריך התקנה ושימוש בדוקר](#)

שאלה 1

בשאלה זו נממש מספר פרדיקטים ב-Prolog שיאפשרו לנו לעבוד עם מספרים שלמים.

1. הגדירו את הפרדיקט `pythagorean/3` המתקיים אם הפרמטרים הם [שלושה פיתגורית](#).

ניתן להניח ששלושת הפרמטרים הינם קשורים (bound), כלומר בעלי ערך קונקרטי בזמן הפעלת הפרדיקט.

לדוגמה:

```
?- pythagorean(3, 4, 5).
```

```
true.
```

```
?- pythagorean(3, 3, 5).
```

```
false.
```

2. הגדירו את הפרדיקט `prime/1` המתקיים כאשר הפרמטר הוא מספר ראשוני (בעל שני מחלקים בדיוק).

ניתן להניח שהפרמטר קשור.

לדוגמה:

```
?- prime(4).
```

```
false.
```

```
?- prime(17).
```

```
true.
```

3. לפי [השערת גולדבך](#), כל מספר זוגי גדול מ-2 הוא סכום של שני מספרים ראשוניים.

הגדירו את הפרדיקט `goldbach/3` המתקיים עם שני הפרמטרים הראשונים הינם מספרים ראשוניים, והשלישי הוא מספר זוגי השווה לסכומם.

ניתן להניח שהפרמטר השלישי (בלבד) קשור.

לדוגמה:

```
?- goldbach(X, Y, 20).
```

```
X = 3, Y = 17 ;
```

```
X = 7, Y = 13 ;
```

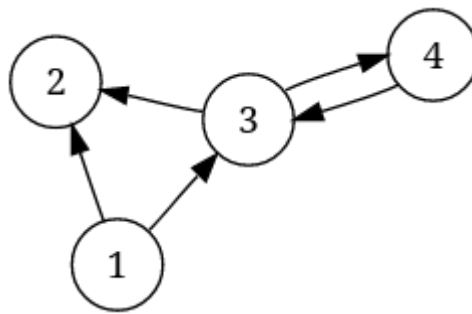
```
X = 13, Y = 7 ;
```

```
X = 17, Y = 3.
```

שאלה 2

לטובת המחקר שלו, המתרגל האחראי בקורס שפות תכנות רצה להשתמש בתכנות לוגי כדי לנתח גרפים מכוונים (מבוסס על סיפור אמיתי). בשאלה זו נממש תכנית פרולוג שתעזור לו בניתוח הגרפים.

גרף בתכנית נתון על ידי הפנקטור graph ורשימה דו-מימדית המייצגת רשימת שכנויות. הרשימה במקום ה-i (החל מ-1) היא רשימת כל הצמתים אליהם יוצאת קשת מהצומת i. לדוגמה, עובר הגרף הבא:



הביטוי שמייצג אותו הוא:

```
graph([
  [2, 3],
  [],
  [2, 4],
  [3],
]).
```

הערה - ניתן להניח שהגרף פשוט, כלומר ללא קשתות עצמיות או קשתות מקבילות.

1. הגדירו את הפרדיקט legal_graph/1 כאשר הפרמטר שלו הוא גרף חוקי. גרף הוא חוקי אם מכיל לפחות צומת אחד ואין בו קשתות לצמתים שאינם בגרף. לדוגמה:

```
?- legal_graph(graph([[2, 3], [], [2, 4], [3]])).
true.
?- legal_graph(graph([])).
false.
```

2. הגדירו את הפרדיקט $\text{edge}/3$ המתקיים כאשר הפרמטר הראשון הוא גרף וקיימת קשת בגרף מהפרמטר השני לשלישי.
לדוגמה:

```
?- edge(graph([[2, 3], [], [2, 4], [3]]), X, 3).
```

```
X = 1 ;
```

```
X = 4.
```

```
?- edge(graph([[2, 3], [], [2, 4], [3]]), 2, X).
```

```
false.
```

3. הגדירו את הפרדיקט $\text{path}/4$ המתקיים כאשר הפרמטר הראשון הוא גרף בו הארגומנט הרביעי הוא מסלול פשוט לא ריק (עם קשת אחת לפחות וחסר מעגלים) המתחיל בפרמטר השני ומסתיים בפרמטר השלישי.
לדוגמה:

```
?- path(graph([[2, 3], [], [2, 4], [3]]), 1, 4, P).
```

```
P = [1, 3, 4].
```

```
?- path(graph([[2, 3], [], [2, 4], [3]]), 2, 4, _).
```

```
false.
```

4. הגדירו את הפרדיקט $\text{circle}/2$ המתקיים אם הפרמטר הראשון הוא גרף בו קיים מעגל שהוא הפרמטר השני.
לדוגמה:

```
?- circle(graph([[2, 3], [], [2, 4], [3]]), P).
```

```
P = [3, 4, 3] ;
```

```
P = [4, 3, 4].
```

5. הגדירו את הפרדיקט $\text{dag}/1$ המתקיים אם הפרמטר הוא גרף בו לא קיימים מעגלים.
לדוגמה:

```
?- dag(graph([[2, 3], [], [2, 4], [3]])).
```

```
false.
```

```
?- dag(graph([[2, 3], [], [2], [3]])).
```

```
true.
```

הנחיות

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:
hw5_q1.pl, hw5_q2.pl, dry.pdf.
- על החלק היבש להיות מוקלד. אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX5_ID1_ID2.zip כאשר ID1, ID2 הם מספרי ת.ז. של המגישים.
- מסופק לכם סקריפט לבדיקת עצמית של קובץ הגשה בשם **selfcheck.sh**. אנא בדקו את תקינות ההגשה שלכם בעזרת הסקריפט בסביבת ה-docker.
- בודקי התרגילים **מאוד** אוהבים Memes. שתפו את תחושותיכם במהלך פתירת התרגיל באמצעות Memes מתאימים ב-pdf של החלק היבש. Memes מצחיקים בצורה יוצאת דופן יזכו את היוצרים בבונוס.

בהצלחה!