

שפות תכנות / חורף תשע"ז

תרגיל בית 1

General Introduction, Pascal, ML Introduction, (E)BNF

תאריך פרסום: 28.3.2017

מועד אחרון להגשה: 18.4.2017

מועד אחרון להגשה מאוחרת: 23.4.2017

מתרגלת אחראית: רנא שחות

אי-מייל: ranas@cs.technion.ac.il

בפניה בדוא"ל, נושא ההודעה (subject) יהיה "PLS2017 EX1" (ללא המרכאות).

תרגיל בית זה מורכב משני חלקים, חלק יבש וחלק רטוב. לפני ההגשה, ודאו שההגשה שלכם תואמת את הנחיות ההגשה שנמצאות [כאן](#).

1. חלק יבש

שאלה 1

עבור כל אחד מהטיפוסים הבאים **בשפת C**, כתבו אם הוא אטומי או לא, ואם הוא first-class או לא, והסבירו את תשובותיכם.

1. int
2. int*
3. void
4. struct with 2 int fields
5. string, represented as a continuous sequence of characters which ends with '\0'
6. enum { true, false }
7. function of int to int

שאלה 2

עבור כל אחד מהטיפוסים הבאים **בשפת ML**, כתבו אם הוא אטומי או לא, ואם הוא first-class או לא, והסבירו את תשובותיכם.

1. int
2. int*int
3. string

4. bool
5. int->int

שאלה 3

סכמו את ההבדלים העיקריים בין BNF ו-EBNF. היעזרו בערכי ה-wikipedia של [BNF](#) ו-[EBNF](#) ובחומר ההרצאות.

שאלה 4

עיינו בדקדוק של שפת Mini-Java המופיע [כאן](#) וענו על הסעיפים הבאים:

- a. מהו סגנון ה-EBNF שבו משתמשים?
 - i. כלומר, כיצד מסומנים ליטרלים (terminals)? כיצד מופרדים כללים? וכו'.
 - b. כתבו תוכנית קצרה ככל הניתן בשפת Mini-Java ושרטטו את עץ הגזירה שלה.
 - c. האם שפת Mini-Java היא ספרטיסטית או טרמיניסטית? הסבירו. (ניתן להיעזר במסמך [צעדים ראשונים](#)).
 - d. כתבו ביטוי רגולרי המתאר את המזהים החוקיים בשפת Mini-Java.

שאלה 5

ענו על הסעיפים הבאים:

- a. שפת פסקל מתיימרת להיות לא תלויה במכונה. זה לא מדויק. מצאו פרטים על המכונה שעליה מומשה פסקל לראשונה, והראו שתכנון השפה הושפע מהמכונה הזו.
רמז: קיים באינטרנט דיון על הסוגייה הזו. אתם מוזמנים למצוא את הדיון ולצטט משם.
- b. הכינו רשימה של כל המילים השמורות בשפה, וסווגו אותן לפי הקריטריונים שמופיעים בפרק "[צעדים ראשונים](#)".

שאלה 6

בכל אחד מהסעיפים הבאים תנו דוגמה לשפה שמקיימת את הדרישה והדגימו את השימוש בכל סוג של הערה בהתאם לסעיף:

דוגמה: "שפה שבה יש הערות שורה וגם הערות בלוק"
תשובה:

- שפת C.

• דוגמה להערת שורה:

```
// This is a comment
```

• דוגמה להערת בלוק:

```
/* This is  
   also a comment */
```

a. שפה שבה יש רק הערות שורה.

b. שפה שבה יש רק הערות בלוק.

c. שפה שבה ניתן לקנן הערות בלוק.

d. שפה שבה יש לפחות שלושה סוגים של הערות שורה.

e. שפה שבה יש הבדל בסינטקס בין הערת שורה שמתחילה מיד בתחילת השורה לבין הערת שורה שמתחילה באמצע השורה.

שאלה 7

הסבירו מדוע הקריאות הבאות מחזירות שגיאה ב ML:

1. `"h" ^ #"C";`
2. `84 / 2;`
3. `fun f x = if x = 0 then x else false;`
4. `fun f x = if x = #"a" then x ^ "b" else x ^ "c";`
5. `5 - (-3);`
6. `Math.sqrt(9);`
7. `sin(3.14);`
8. `val if = 5;`
9. `String.sub("hello", 5);`
10. `fun sqrt_of_int x:int = Math.sqrt(real(x));`

2. חלק רטוב

2.1 פסקל

שימו לב: גרסת פסקל שבה עליכם להשתמש היא 2.6.4 כפי שכתוב [באתר](#), ולא כפי שכתוב במצגת שהועברה בתרגול הראשון.

נתון ה-quine הבא בשפת Pascal:

```
Program Dolly;
const
  prelines = 9;
  numlines = 23;
var
  myself : array [1..numlines] of string[100];
  i, j : integer;

begin
  myself[1] := 'Program Dolly;';
  myself[2] := 'const';
  myself[3] := '  prelines = 9;';
  myself[4] := '  numlines = 23;';
  myself[5] := 'var';
  myself[6] := '  myself : array [1..numlines] of string[100];';
  myself[7] := '  i, j : integer;';
  myself[8] := '';
  myself[9] := 'begin';
  myself[10] := '';
  myself[11] := '  for i := 1 to prelines do writeln(myself[i]);';
  myself[12] := '  for i := 1 to numlines do';
  myself[13] := '    begin';
  myself[14] := '      write(''  myself['', i:1, ''] := '');';
  myself[15] := '      for j := 1 to length(myself[i]) do';
  myself[16] := '        begin';
  myself[17] := '          if myself[i,j] = '''''''' then write(''''''');';
  myself[18] := '        write(myself[i,j]);';
  myself[19] := '      end;';
  myself[20] := '    writeln(''''''');';
  myself[21] := '  end;';
  myself[22] := '  for i := prelines+1 to numlines do writeln(myself[i]);';
  myself[23] := 'end.';

  for i := 1 to prelines do writeln(myself[i]);
  for i := 1 to numlines do
    begin
      write('  myself['', i:1, ''] := '');
      for j := 1 to length(myself[i]) do
        begin
          if myself[i,j] = '''' then write(''');

```

```

        write(myself[i,j]);
    end;
    writeln('');
end;
for i := prelines+1 to numlines do writeln(myself[i]);
end.

```

ודאו שברור לכם מדוע התוכנית הזו אכן מדפיסה את הקוד של עצמה.

כעת, שנו את התכנית כך שהיא תדפיס את הקוד מקור של עצמה **פעמיים**, במקום פעם אחת. שימרו את הקובץ תחת השם **dubquine.pas**.

בשביל לעזור לכם להבין כיצד כותבים quines באופן כללי, אתם יכולים להיעזר במצגת הזו שעברה בתרגול הראשון בסמסטר הקודם או בעמוד הויקיפדיה המתאים.

תרגיל זה הוא התרגיל היחיד שבו אתם לא חייבים להוסיף הערה עם שמות המגשים כפי שנדרש בהנחיות ההגשה.

שימו לב: אם בכל זאת החלטתם להוסיף שורת הערה בתחילת הקוד, עליכם לוודא שהתוכנית מדפיסה את עצמה פעמיים עם ההערה! (אחרת, לא מתקבל quine כפול, ולא עניתם נכון על השאלה).

אופן הבדיקה של הסעיף הזה

בעקבות פניות רבות, להלן תיאור מדויק של הדרך שבה ייבדק הסעיף הזה:

- נסמן את הקוד שתכתבו ב-CODE
- נסמן את הפלט של הקוד שלכם ב-OUTPUT
- להלן הבדיקה:

$OUTPUT == CODE \wedge CODE$

כאשר האופרטור " \wedge " הוא אופרטור השרשור.

שימו לב: אם בקוד שלכם מופיעה ירידת שורה, היא צריכה להופיע בפלט. אותו דבר לגבי רווחים, טאבים וכל שאר התווים שמופיעים בקוד. הבדיקה שלעיל היא ברורה מאוד: אם הקוד שתכתבו לא מדפיס את התוכנית שכתבתם פעמיים (עד לתו האחרון), הבדיקה תיכשל. זו המשמעות של quine (או qubquine במקרה שלנו).

ML 2.2

למען הסר ספק:

- אין מניעה להגדיר ולהשתמש בפונקציות עזר
- מתוך ההנחיות שבסוף התרגיל:
 - בתרגיל זה ניתן להשתמש רק בחומר שנלמד עד **תרגול 2 (כולל)**. אין להשתמש באף פונקציה/תכונה של השפה שלא נלמדה בתרגולים.

פלינדרום

פְּלִינְדְרוֹם הוא מילה, מספר, משפט או כל רצף סמלים אחר שקריאתו מימין לשמאל ומשמאל לימין היא זהה (כאשר מתעלמים בדרך כלל מסימני ניקוד, פיסוק ותווים מיוחדים אחרים). מקור המילה ביוונית: *πάλιν* (פָּלִין - "שוב", "הפוך") ו-*δρομος* (דרומוס - "כיוון").

<https://he.wikipedia.org/wiki/%D7%A4%D7%9C%D7%99%D7%A0%D7%93%D7%A8%D7%95%D7%9D>

ממשו את הפונקציות הבאות:

1. כתבו פונקציה שבודקת האם מחרוזת הקלט היא פלינדרום

```
isPalindrome: string -> bool
```

הפונקציה תקבל מחרוזת ותחזיר true אם המחרוזת היא פלינדרום אחרת, הפונקציה תחזיר false

דוגמאות הרצה:

```
- isPalindrome("level");  
val it = true : bool
```

2.

```
countLetter : string * char -> int
```

הפונקציה תקבל מחרוזת ואות ותחזיר כמה פעמים האות מופיעה במחרוזת.

אם האות לא מופיעה במחרוזת הפונקציה אמורה להחזיר 0

דוגמאות הרצה:

```
- countLetter("upper", # "p");  
val it = 2 : int
```

3.

```
nonRepeatedLetter : string -> char
```

הפונקציה תקבל מחרוזת ותחזיר את האות הראשונה שלא חוזרת על עצמה במחרוזת.

הנחה מקלה: הפונקציה תיבדק אך ורק על מחרוזת שבהן קיימת אות כלשהי שאינה חוזרת על עצמה.

דוגמאות הרצה:

```
- nonRepeatedLetter("helloworld");  
val it = #"h" : char
```

הנחיות הגשה

כללי

- את התרגיל יש להגיש בזוגות.
- תרגיל אשר הוגש על ידי קבוצה בגודל שונה מ-2 ללא אישור של המתרגל האחראי לא ייבדק.
- יש להקפיד על קוד קצר, קריא ויעיל עד כמה שניתן. הקפידו על הזחות (indentations), אחידות וקריאות.
- יש להקפיד לבדוק את הקוד.
- תיקונים והבהרות לתרגיל יתווספו לתרגיל בצבע כתום ויתועדו כאן.
- שימו לב שהתיקונים וההבהרות לתרגיל הם מחייבים, ולא יתקבלו דחיות או ערעורים עקב אי קריאת החלק הזה.
- אין להגיש קוד עם שגיאות. יש לוודא כי הקובץ אותו אתם מגישים נטען ללא שגיאות ורץ בצורה המצופה.
- בתרגיל זה, כמו בכל התרגילים, יש לכתוב קוד לפי המהדורה המקורית של פסקל. אין להשתמש בהרחבות של השפה שקיימות ב-freepascal או בכל מהדר אחר.
- שימו לב כי ML היא case-sensitive, ודאו על כן כי שמות הפונקציות הן בדיוק כפי שנדרש בתרגיל.
- בתרגיל זה ניתן להשתמש רק בחומר שנלמד עד תרגול 2 (כולל). אין להשתמש באף פונקציה/תכונה של השפה שלא נלמדה בתרגולים.

הוראות הגשה "יבשה"

- את החלק היבש יש להגיש לתא הקורס בקומה 1 ובנוסף לצרפו להגשה האלקטרונית באתר הקורס (כפי שמוסבר בהוראות ההגשה האלקטרונית).
- החלק היבש כולל:
- דף שער פשוט המכיל את מספר הקורס, פרטי המגישים (שם, ת"ז, וכתובת דוא"ל), ואת מספר התרגיל בית.
 - את תשובותיכם לחלק היבש בלבד. אין צורך להדפיס את הקוד.
 - על מנת לחסוך בדפים, אנא הדפיסו משני צדי הדף.

הוראות הגשה אלקטרונית

ההגשה האלקטרונית תתבצע באתר הקורס ותכיל את הקבצים הבאים:

- הקובץ dry.pdf הכולל את התשובות לחלק היבש של התרגיל.
- הקובץ dubquine.pas הכולל את הקוד של חלק 2.1.
- הקובץ ex1.sml הכולל את הפונקציות שנדרשתם לממש בחלק 2.2.

לכל קובץ שמכיל קוד:

- הוסיפו בשורה הראשונה הערה המכילה את השם, מספר ת.ז. וכתובת המייל של המגישים מופרדים באמצעות רווח.
- וודאו שהקבצים נטענים/מתקמפלים ללא שגיאות גם לאחר הוספת ההערה שלעיל.

יש להגיש את הקבצים דחוסים בתוך קובץ zip. הקבצים יהיו בשורש קובץ ה-zip ולא בתוך ספרייה. שם הקובץ יהיה EX1_ID1_ID2.zip כאשר ID1, ID2 הם מספרי ת.ז. של המגישים. שימו לב שהבדיקה של החלק הרטוב היא אוטומטית, ולכן הקפידו על מילוי כל ההוראות בשביל למנוע בעיות מיותרות.

הוראות הגשה מאוחרת

בהתאם לדף הנהלים, ניתן להגיש את התרגיל באיחור של עד 72 שעות בתמורה לקנס של 5 נקודות. להזכירכם, הסיבה היחידה שבגינה ניתן לקבל דחייה בהגשת תרגיל הבית היא מילואים. על כל יום מילואים של אחד מהשותפים, תקבלו דחייה של יום אחד בהגשת תרגיל הבית. תרגילים שיוגשו באיחור של מעל 72 שעות (ללא אישור מהמתרגל האחראי על התרגיל), לא ייבדקו.

שימו לב:

- ההגשה היבשה תתבצע ישירות לתא של המתרגל האחראי על התרגיל בקומה 5, ולא לתא הקורס בקומה 1.
- ההגשה האלקטרונית תישאר פתוחה עד 72 שעות לאחר תאריך ההגשה של התרגיל על מנת לאפשר למגישים באיחור להגיש אלקטרונית את התרגיל. אין להגיש את התרגיל ישירות למייל של המתרגל האחראי על התרגיל.

בהצלחה!