שפות תכנות 234319 - תרגיל בית 1

חלק יבש

1. מה ההבדל בין המושג "Natural Language Programming", ובין המושג "Naturalistic Programming"?

Naturalistic Programming היא שימוש בתכונות משפות "טבעיות" כמו עברית, כדי להביע את רצון המתכנת בצורה מיטבית(מנקודת מבטו של המתכנת),
לעומת זאת Natural Language Programming זאת צורה להגדיר שפת תכנות כשפה עם אלמנטים טביעים, כמו משפטים בשפות פרוצדורליות(C,Pascal).

2. ההבדל בין המושג "קריאת זנב" לבין המושג "רקורסיית זנב".
<u>קריאת זנב</u> - סוג של תהליך/פעולה הנדרשת לסיום התהליך המקורי/סיום האלגוריתם.

<u>רקורסיית זנב</u> - זאת פונקציה רקורסיבית המתוכננת כך שתאפשר ריצה בסיבוכיות מקום נכונה יותר מאשר סיבוכיות מקום פרופורציונלית לעומק מחסנית הרקורסיה. בשיטה זאת בסוף ריצת האלגוריתם אין צורך לבצע "התקפלות" לאחר שלב אחר שלב, כלומר רקורסיה מסוג זה לא שומר את כתובות החזרה לפונקציות הקודמות, אלה רק לפונקציית המקור שקראה לאותה הרקורסיה בתחילה.
רקורסיית זנב זה מקרה פרטי של קריאת זנב, כלומר פעולה המתבצעת בסופו של הפונקציה לטובת סיום האלגוריתם הריקורסיבי.

3. נסווג את האופרטורים בפסקל לקבוצות.

סוגי משתנים	קבוצות	מחרוזות	אריתמטיים	יחסים	בוליאנים	פעולות על ביטים
is : בדיקת סוג הטיפוס	+ : איחוד שתי קבוצות	+ : שירשור מחרוזות	+ : חיבור בין שני אופרנדים	= : השוואה	and : בודק אם שני האופרנדים אמת	and : פעולת וגם
as : התנאי של טיפוסיות	- : חיסור קבוצות/ הבדלים בינהם		- : חיסור בין שני אופרנדים	<> : לא שווה	xor : פעולה לוגית של xor בין שני אופרנדים.	or: פעולת או
	* : חיתוך בין קבוצות		* : כפל בין שני אופרנדים	> : גדול	or : אחד מהאופרנדים מתקיים	not : פעולת היפוך
	>< : הפרש סימטרי בין קבוצות		/ : חילוק בין שני אופרנדים	< : קטן	not : היפוך, על אופרנד יחיד	xor : זוגיות האחדות
	<= : הכלה של קבוצות		mod : שארית בין אופרנדים	>= : גדול שווה		shr,>> : הזזת ביטים ימינה כגודל האופרנד ה2
	include : להוסיף איבר לקבוצה		div : חלוקה שלמה בין אופרנדים	<= : קטן שווה		shl,<< : הזזת ביטים שמאלה כגודל האופרנד ה2
	exclude : להוציא איבר מהקבוצה		+ : סימן האופרנד	in : אותו דבר כמו בקבוצות(אותו אחד)		
	in : לבדוק שייכות לקבוצה		- : היפוך הסימן			

4. ערכו של משתנה שלא אותחל, כיצד ניתן לאתחל משתנה בזמן ההצהרה, וכיצד ניתן באמצעות הפקודה Default.

כברירת מחדל המשתנים בפקסל לא מאותחלים,כלומר בדומה לשפת C הם מכילים זבל.
על מנת לאתחל משתנה בהכרזה נשתמש ב"=". לדוגמא: $var\ x:\ integer\ =\ 15;$
פקודת Default מאתחלת את המשתנה לערך ה"אפס" של הטיפוס, לדוגמא: integer מאותחל לאפס.

5. דוגמא לטיפוסיות חזקה בפסקל

$var\ a : integer = 5;$
$var\ b : real = 0;$
$b := a + 1;$

לא יתקמפל בגלל הטיפוסיות החזקה בפסקל.
אין השמה של integer לתוך real.

6. הבעייתיות במערכים בפסקל
בגלל הטיפוסיות החזקה של פסקל, גודל המערך מהווה הגדרה המבדילה בין שני מערכים בגדלים שונים(אף אם הם מערכים של אותו הטיפוס).
לדוגמא אין אפשרות לרשום פונקציה גנרית לאיתחול מערך בלי תלות בגודלו.
- בשפת פסקל: לא ניתן.
- בשפת C:

$$
\begin{aligned}
&void\ easyArrInit(int * arr, int\ n)\{\\
&\qquad for(int\ i = 0; i < n; i++)\\
&\qquad\qquad arr[i] = 0;\\
&\}
\end{aligned}
$$

7. המגבלה על גודלו של set בשפת פסקל, ומגבלה על טיפוס היעד
יכול להחזיק עד 256 איברים בset יחיד.
המגבלה על טיפוס היעד היא שset יכול רק להחזיק טיפוסים פשוטים(לדוגמא לא מערכים).

8. כיצד בונים מהדר שמהדר את עצמו
תהליך זה נקרא bootstrapping,
כלומר כתיבת מהדר לשפה X באחת משתי האפשרויות הבאות:
- כתיבת המהדר בשפה Y ולאחר מכן ע"י המהדר הנ"ל לבנות מהדר חדש בשפה X.
- כתיבת המהדר בשפה X והידורו באמצעות מהדר ישן יותר, כלומר עדכון גרסת המהדר.

9. כתבו EBNF המגדיר את ליטרל המספרים

```
expretion = (number, math, number) | (number);
number = ([negative], {digits}, [dot], digits, {digits}) | ([negative], digits ,{digits}, [dot],
{digits});
signs = {dot , negative , math};
mid = {digits}
digits = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
dot = ".";
negative = “-" | “+”;
math = ”E" | “e”;
```

ביטוי בRegExp:

```
/([\+\-]?((\d+[\.]?\d*)|([\.]?\d+)))([Ee]?[\+\-]?((\d+[\.]?\d*)|([\.]?\d+)))?/g
```