

חלק יבש – גיליון 4

1) להלן הרחבת התוכנית הראשונה שכתבנו בפסקל. נעיר שכיוון שבכל לולאה יש לנו תנאי בודד אז מספר התנאים שקול למספר האיטרציות ולכן אנו מדפיסים את אותו מונה. נקמפל זאת בלינוקס ע"י הפקודה `pc` ושם הקובץ, לאחר מתן הרשאות נוכל להריץ ולראות תוצאה.

```
program TaxicabNumber;
  var count,k,a,temp : integer;
  function qube(num : integer) : integer;
    begin
      qube := num*num*num;
    end;
  function isTwoQubes(k,a : integer) : boolean;
    begin
      if round(exp(1/3*ln(k-qube(a))))- exp(1/3*ln(k-qube(a))) >= 0 then
        isTwoQubes := round(exp(1/3*ln(k-qube(a))))- exp(1/3*ln(k-
qube(a))) <= 0.00000001
      else
        isTwoQubes := exp(1/3*ln(k-qube(a))) - round(exp(1/3*ln(k-
qube(a)))) <= 0.00000001;
      end;
  function isTaxiNumber(var k,a,count : integer) : boolean;
    begin
      isTaxiNumber := false;
      while qube(a)<(k/2) do
        begin
          count := count+1;
          if isTwoQubes(k,a) then
            begin
              isTaxiNumber := true;
              break;
            end;
          a := a+1;
        end;
      end;
    begin
      k := 3;
      count := 0;
      while true=true do
        begin
          count := count+1;
          a := 1;
```

```

                                if isTaxiNumber(k,a,count) then
                                    begin
                                        temp := a;
                                        a := a+1;
                                        if isTaxiNumber(k,a,count) then
                                            begin
                                                if (qube(temp)+qube(a)) <> k
then
                                                    begin
                                                        WriteLn('The
taxicab number is ',k);
                                                        WriteLn('The
counter is ',count);
                                                        WriteLn('The
number of conditions are ',count);
                                                    break;
                                                    end;
                                                end;
                                            end;
                                        k := k+1;
                                    end;
                                end.

```

2) תרגום של התוכנית בפסקל לפייטון (באופן דומה עד כמה שניתן):
קימפלנו בלינוקס ע"י הפקודה python ושם הקובץ, מתן הרשאות והרצה.

```

def cube(num):
    return num*num*num
def cube_root(num):
    return num**(1./3.)
def is_two_cubes(k, a):
    qr = cube_root(k-cube(a))
    return abs(round(qr)-qr) <= 0.00000001

def is_taxi_number(k):
    global count
    global a
    while cube(a) < (k/2):
        count += 1
        if is_two_cubes(k, a):
            return True
        a += 1
    return False

k = 3
count = 0
temp = 0
while True:
    count += 1
    a = 1
    if is_taxi_number(k):

```

```

temp = a
a += 1
if is_taxi_number(k):
    if (cube(temp)+cube(a)) != k:
        print('the taxicab number is ', k)
        print('the counter is ', count)
        print('the number of conditions are', count)
        break

k += 1

```

הורדתי

(3

גירסא 7.2.3 של swi-prolog על מע' הפעלה ווינדוס וקימפלתי את התוכנית
הבאה במפרש:

```

1 ?- between(4,12,I),
|   between(4,12,L),
|   plus(I,D,L),
|   between(1,2,D),
|   plus(I,L,S),
|   between(0,19,S),
|   format('my age is ~w , while lily age is ~w',[I,L]).

```

4) בשפת נים יש שימוש בהעברה by value וגם by reference , העברת טיפוסים פרמטיבית נעשית ע"י העברת ערך למעט מקרים בהם הקומפיילר מחליט מטעמי יעילות ובטיחות שניתן להעביר ע"י הפנייה
var parameters are never necessary for efficient parameter passing. Since non-var parameters cannot be modified the compiler is always free to pass arguments by reference if it considers it can speed up execution.

5. ב-NIM יש 3 מתוך 5 סוגי מערכים:

1) ישנם מערכים קבועים שנמצאים על המחסנית (אך גודלם יכול להקבע בזמן ריצה):

"Often fixed size arrays turn out to be too inflexible; procedures should be able to deal with arrays of different sizes. The openarray type allows this. Openarrays are always indexed with an `int` starting at position 0."

2) ישנם מערכים גמישים שנמצאים על הערימה:

"Sequences are similar to arrays but of dynamic length which may change during runtime (like strings). Since sequences are resizable they are always allocated on the heap and garbage collected."

3) ישנם מערכים סטטיים כמו ב-C שגודלם נקבע בקומפילציה:

"An array is a simple fixed length container. Each element in the array has the same type. The array's index type can be any ordinal type."

6) נכתוב סקריפט באש בשם `bscript.sh`, ונריץ `bscript.sh`.

```
#!/bin/bash
```

```
cat bscript.sh
```

7) void safety היא התכונה בשפות תכנות מונחה עצמים המונעת את האפשרות שאובייקטים יהיו בעל ערך null, ובכך מונעת אפשרות לקבל חריגת null exception בזמן ריצה-ולכך בדר"כ לקריסת התוכנית(במידה ולא מטופלת). ++C מציע פתרון חלקי בנושא, וזה ע"י שימוש ב reference ים (&) שהם למעשה מצביעים שלא יכולים להכיל null. נים, בדומה לגאווה, מאפשרת להתמודד גם כן באופן חלקי ע"י סימון הפניה שלא יכולה להכיל nil (null) ע"י האנוטציה `not nil`. דוגמה:

```
type
  PObject = ref TObj not nil
  TProc = (proc (x, y: int)) not nil

proc p(x: PObject) =
  echo "not nil"

# compiler catches this:
p(nil)

# and also this:
```

```
var x: PObject  
p(x)
```

בנוסף open arrays לא יכולים לקבל NULL.

8) שפות שמצאתי כי על שם נשים:

Ada – על שם [Ada Lovelace](#)

Linda – על שם שחקנית הפורנו [Linda Lovelace](#)

הסיבה לכך שיש שפה על שמה נאמר במכתב הבא שפורסם בNew York Times:

from the 1/19/92 New York Times, "[DavidGelernter](#)'s Romance with Linda" (sect 3, p1,6)

When it came time to name the language, Mr Gelernter said he noted that Ada was named after Ada Augusta Lovelace, the daughter of Lord Byron, the English poet. Miss Lovelace is regarded as the first computer programmer because she worked for the computer pioneer Charles Babbage.

Another woman named Lovelace was in the news when Mr Gelernter was casting about for a name -- Linda Lovelace, a star of pornographic films. So he named the language Linda, and it stuck. Asked about it now, Mr Gelernter grins and shrugs, "I was a graduate student at the time," he said.

מקור: <http://spaf.cerias.purdue.edu/Yucks/V2/msg00008.html>

Brenda – (אם כי זה שייך לספרות זהו שם של אישה).

Alice

Miranda

(9 חלק יבש על מלכי ישראל:

על מנת לפתור בעיה זו חילקתי את הבעיה לתת משימות הבאות:

דף הויקיפדיה פורסס לפורמט הבא:

<startYear> ,<endYear>,<kingName>,<house> ,

קובץ זה נשמר על הלינוקס בשם kings ותוכנו:

```
931,909,Jeroboam I,Jeroboam I,
909,908,Nadab,Jeroboam I,
908,885,Baasha,Baasha,
885,884,Elah,Baasha,
844,844,Zimri,Zimri,
884,873,Omri,Omri,
873,852,Ahab,Omri,
852,851,Ahaziah,Omri,
851,842,Joram,Omri,
842,815,Jehu,Jehu,
819,804,Jehoahaz,Jehu,
805,790,Jehoash,Jehu,
790,750,Jeroboam II,Jehu,
750,749,ÊZachariah,Jehu,
749,749,Shallum,Shallum,
749,738,Menahem,Menahem,
738,736,Pekahiah,Menahem,
736,732,Pekah,Pekah,
732,722,Hoshea,Hoshea,
```

לאחר מכן בניתי סקריפט בAWK שבונה תוכנית בפסקל ו"שותל" את הפרטים ברשומות בהתאם לקובץ kings, ולבסוף ע"מ לאפשר הפעלה פשוטה, סביבה נקייה ושינויים נוספים הנדרשו בסקריפט AWK בניתי סקריפט בבאש שדואג להריץ את הסקריפט AWK, מקמפל את הקובצי פסקל שיצר, מריץ אותם ולבסוף מוחק את הקבצים המקומפלים והקבצי ריצה.

לסיכום: יש להריץ את הסקריפט ./compile_and_run. המייצר את קבצי ההרצה השונים הכותבים ל kings.dat, עם הרצת B לסירוגין והדפסת סיכום של פלט ושגיאות.

הסקריפט BASH העונה לשם compile_and_run:

```
#!/bin/bash
cat kings | ./readAndCreateAInPascal > A.pas
fpc A.pas
./A
rm A.o
rm A
fpc B.pas
./B > output
echo "after reading from A" >>output
rm B.o
cat readAndCreateAInPascal | sed -re 's/name/othername/g' | sed -re 's/startYear/otherStartYear/g' |
sed -re 's/endYear/otherEndYear/g' > readAndCreateCInPascal
chmod 777 readAndCreateCInPascal
cat kings | ./readAndCreateCInPascal > C.pas
fpc C.pas
./C
./B >> output 2>errors.C
echo "after reading from C" >>output

rm C.o
rm C
#building D
#singlequote=\047
cat readAndCreateAInPascal | sed -re 's/BCE=integer;/BCE=integer;
crowneChoice=(rebellion,inheritance);/g' | sed -re 's/endYear:BCE;/endYear:BCE;
kingBy:crowneChoice;/g' | sed '23 a print "if kings["i"].name=" "\047"$4"\047" " then";' | sed '24 a
print " kings["i"].kingBy:=rebellion";' | sed '25 a print "else";' | sed '26 a print "
kings["i"].kingBy:=inheritance";' > readAndCreateDInPascal
```

```
chmod 777 readAndCreateDInPascal
cat kings | ./readAndCreateDInPascal > D.pas
fpc D.pas
./D
./B >> output 2>errors.D
echo "after reading from D" >>output
rm D.o
rm D

cat readAndCreateDInPascal | sed -re 's/name:string;/name:string;father:string;/g' >
readAndCreateEInPascal
chmod 777 readAndCreateEInPascal
cat kings | ./readAndCreateEInPascal > E.pas
fpc E.pas
./E
clear
./B >>output 2>errors.E
echo "after reading from E" >>output
echo "-----All programs output-----"
cat output
echo "-----all programs errors-----"
cat errors.*
rm E.o
echo "-----Done-----"
rm E
rm B
```



```
#!/usr/bin/awk -f

BEGIN{
FS=",";
print "program A;";
print "type";
print "BCE=integer;";
print "King= record";
print"        name:string;";
print"        startYear:BCE;";
print"        endYear:BCE;";
print "end;";
print"var";
print "f:file of King;";
print "kings :array [0..18] of King;";
print "i:integer;";
print "BEGIN";
i=0;
}

#awk body
{
print "kings["i"].name:=" ""$3"";";
print "kings["i"].startYear:=" $1"";";
print "kings["i"].endYear:=" $2"";";
i++;
}

END{
print " (* writing kings to kings.dat file *)";
print"        Assign(f,'kings.dat');";
```

```
print"          (* opens file for writing *)"  
print"          Rewrite(f);"  
print"          (* write king records *)";  
print"  for i:=0 to 18 do Write(f,kings[i]);"  
print"(* closing file *)";  
print"  Close(f);"  
print"  END.";   
}
```

```
program B;

type
  BCE=integer;
  King= record
      name:string;
      startYear:BCE;
      endYear:BCE;
  end;

var
  kingvar : King;
  f:file of King;
  maxBCE:BCE;
  maxName:string;
  diff:BCE;
BEGIN
  maxBCE:=0;
  diff:=0;
  maxName:='Unkown';
  assign(f, 'kings.dat');
  (* open a file for reading*)
  reset(f);
  while not eof(f) do
  begin
    read(f,kingvar);

    diff:=kingvar.startYear-kingvar.endYear;
    if( diff>maxBCE) then
```

```

begin
    maxBCE:=diff;
    maxName:= kingvar.name;

end;

end;

writeln('name of the longest kingsment is ', maxName);

close(f);

END.

```

h. ידוע לנו מהשיעור כי פסקל אמורה להיות שפה בעלת שקילות שמית, ולכן החל מתוכנית C (והלאה) וקריאתה מהקובץ kings.dat (לסירוגין עם תוכנית B) היה מצופה בתרגיל כי תחול שגיאה, אך מכיוון שבשפה וסביבה שהרצנו (שהורדה מהאתר הקורס) אין אכיפה לדברים אלו (שקילות שמית ומבנית), נוצר type punning והתוכנית עוברת ללא שום שגיאת זמן ריצה אלא פשוט קוראת תוכן שאינו נכון, ובכך מבתצעת שגיאה לוגית. ע"מ לגרום לשגיאת זמן ריצה היה על התרגיל להכיל דרישה לגישה לשדה עם מצביע. שגיאה לוגית זו היתה גוררת שגיאת זמן ריצה, שהרי אז סיכוי גדול שהיינו נגשים לתא זכרון שאינו שייך לנו והמע' הפעלה היתה זורקת אותנו (חשוב להדגיש כי אין שום קשר לאכיפה של השפה בנושא).

אך בתרגיל זה לא היתה דרישה כזו, ולכן התוכנית נקייה משגיאות.

פלט התוכנית עם סיכום השגיאות:

```

-----All programs output-----
name of the longest kingsment is Jeroboam II
after reading from A
name of the longest kingsment is Jeroboam II
after reading from C
name of the longest kingsment is Jeroboam I
after reading from D
name of the longest kingsment is
after reading from E
-----all programs errors-----
-----Done-----
ashiber@ubuntu:~/programingLang$

```

הוכחה למאמץ הרב שהושקע בתרגיל ולתיעוד השיחה עם יוסי גיל המרצה:

https://www.facebook.com/groups/programming.languages/permalink/1095516073824904/?comment_id=1097784186931426&comment_tracking=%7B%22tn%22%3A%22R1%22%7D

חיבורן על שימוש ב file handles בפקל מודרני (לא רואה טעם להעתיק משם, מאוד מתומצת שם וברור)

http://www.tutorialspoint.com/pascal/pascal_files_handling.htm

יכלתי לגרום לשגיאת זמן ריצה בדרך הבאה:

נגדיר שדה יחס (ratio), שהוא יחס השנים בין התחלת מלכות לסוף. בעקבות פיגעת מבנה הרשומה בזמן קריאה נקבל שגיאת זמן ריצה על חלוקה באפס. והרי דרך נוספת לגרום לשגיאה לוגית להיות שגיאת זמן ריצה.