

سلسلة من المقالات و الأبحاث العلمية المكتوبة  
باللغة العربية تبدأ من اليوم 2018/12/25 و  
تنتهي بنشر 150 إصدارا بتاريخ 2019/12/25.  
نأمل منكم المساهمة في نشر هذه المقالات  
لدعم المحتوى العربي على شبكة الإنترنت  
ولدفعنا لتقديم المزيد.

# البرولوج

## الجزء الثالث

مجدي عوض

---

بسم الله الرحمن الرحيم ((قل إن صلاتي و نسكي و محياي و مماتي لله رب العالمين)) صدق الله العظيم

## مقدمة

استكمالا لسلسلة "تعلم لغة البرولوج" التي بدأت بتاريخ 2019/01/02 سوف أعمل في هذه الورقة على استكمال دراسة القوائم بلغة البرولوج حيث توقفنا على كيفية تحديد طول القوائم في الجزء السابق. ثم سنقوم بإنهاء هذه السلسلة من خلال تطبيق عدد من الأمثلة.

## القوائم بلغة البرولوج Lists

كنت في الورقة السابقة قد تحدثت عن كيفية كتابة القوائم و انتقلت لتعريف الـ Member و كيفية كتابته و التعامل معه ثم تحدثت عن كيفية كتابة نص برمجي قادر على تحديد طول القوائم و كنت قد أشرت إلى أن القوائم بلغة البرولوج تتألف من رأس و ذيل و أنهيت حديثي على القوائم في الورقة الماضية عبر كتابة نص برمجي قادر على حل لغز آينشتان.

سوف أستكمل في هذه الورقة بحديثي عن القوائم و سوف أستله بكتابة برنامج يعمل على دمج قائمتين معا:

concatenation([],L,L).

concatenation([X1|L1],L2,[X1|L3]):-concatenation(L1,L2,L3).

إن وظيفة النص البرمجي السابق المكتوب بلغة البرولوج هي الدمج بين قائمتين على أن تكون X1 هي رأس القائمة الأولى و L1 ذيل القائمة الأولى بينما تمثل L2 القائمة الثانية و تمثل L3 القائمة الناتجة عن عملية الدمج. بعد كتابة النص و تشغيله يمكنني سؤال البرنامج كما يلي:

?- concatenation([a,b,c],[d,e,f],List).

و ستكون الإجابة كما يلي:

List = [a,b,c,d,e,f]

الآن سوف نتقل لنكتب برنامجا قادرا على تحويل الأرقام إلى حروف و العكس كذلك. ولكن و قبل القيام بذلك أود الإشارة إلى أنني لن أقوم باستخدام تقنية، أسلوب، صيغ، أو طريقة كتابة لم يسبق لي شرحها و أن كل ما سأقوم باستخدامه لكتابة هذا النص البرمجي كنت قد أعطيته نصيبه من الشرح و أن القضية تعتمد على طريقة التفكير و معرفة استنتاج لغة البرولوج إن جاز التعبير. لكتابة هذا البرنامج سوف أبدأ بتعريف الحقائق التالية:

interm(0,zero).

interm(1,one).

interm(2,two).

interm(3,three).

interm(4,four).

interm(5,five).

interm(6,six).

interm(7,seven).

interm(8,eight).

interm(9,nine).

و الآن سأخبر البرولوج أنه إن كانت القائمة المدخلة فارغة فإن القائمة المخرجة سوف تكون فارغة و لذلك أكتب:

inwords([],[]).

و الآن سوف أقوم بتقسيم القائمة الأولى إلى رأس و ذيل على أن يكون رأس القائمة الأولى هو X و الذيل TAIL و سوف تكون T رأس القائمة المخرجة أو الناتجة بينما Z ستمثل ذيلها بعد ذلك سأطلب من البرولوج أن يترجم الأرقام إلى أحرف و الأحرف إلى أرقام كما يلي:

inwords([X|TAIL],[T|Z]):-interm(X,T),inwords(TAIL,Z).

و بهذا أكون قد انتهيت من كتابة النص البرمجي لهذا البرنامج و سوف تجده في المراجع في حال أردت تحميله و تجربته ولكني أكرر بأن ما من داع للقيام بذلك فإمكانك كتابة هذا البرنامج بطريقتك الخاصة ربما بأسلوب أو منطق مختلف تماما عما عملت به.

بالمنطق نفسه و بالتقنيات ذاتها سوف أقوم الآن بكتابة نص برمجي قادر على حذف عنصر في القائمة و على هنا أن أتخيل 3 حالات هي:

1. العنصر الذي أرغب بحذفه هو العنصر الوحيد في القائمة
2. العنصر الذي أرغب بحذفه هو رأس القائمة
3. العنصر الذي أرغب بحذفه هو أحد العناصر التي تشكل ذيل القائمة

و بناء على ذلك أقوم بكتابة السطر الأول في برنامجي بحيث أقول للبرولوج أنني أرغب بحذف Y من قائمة لا تحمل سوى عنصرا واحدا هو Y فإن النتيجة ستكون عبارة عن قائمة فارغة:

del(Y,[Y],[]).

في السطر التالي سوف أقول للبرولوج أنني أرغب بحذف X من قائمة تشكل X رأسها و LIST1 ذيلها فإن النتيجة ستكون LIST1 أي أنني أقول للبرولوج أن يقوم بإظهار ذيل القائمة فقط:

del(X,[X|LIST1],LIST1).

أخيرا و من خلال السطر البرمجي التالي فأنا أخبر البرولوج أنني أريد حذف X من قائمة رأسها Y و ذيلها LIST أي أن رأس القائمة ليس العنصر الذي أرغب بحذفه:

$\text{del}(X,[Y|LIST],[Y|LIST1]) :- \text{del}(X,LIST,LIST1).$

و بهذا أكون قد انتهيت و يمكنك تحميل النص البرمجي من خلال الرابط الخاص به في المراجع أما أنا فسأبدأ بكتابة البرنامج التالي و الذي سوف يعمل على إيجاد كافة الاحتمالات الممكنة لاستبدال مواقع عناصر القائمة (درس الاحتمالات في مادة الرياضيات في المرحلة الثانوية). سوف أستعين بداية بالنص البرمجي السابق لإتمام عمليات الحذف اللازمة في هذا البرنامج.

$\text{del}(X,[X|LIST1],LIST1).$

$\text{del}(X,[Y|LIST],[Y|LIST1]) :- \text{del}(X,LIST,LIST1).$

ثم سأقول للبرولوج أنه في حالة كانت القائمة المدخلة فارغة فإن القائمة الناتجة عن العملية سوف تكون فارغة و أعبر عن هذا بلغة البرولوج من خلال كتابة السطر البرمجي التالي:

$\text{list\_permutation}([],[]).$

و الآن X هي رأس القائمة و P هي ذيل القائمة لذلك يجب أن نقوم بحذف X من القائمة L لنحصل على L1 ثم نقوم بمخاطبة list\_permutation للحصول على P من L1 و هكذا نكون قد قمنا بعملية الاستبدال. كل ما سبق يمكن التعبير عنه بلغة البرولوج من خلال السطر البرمجي التالي:

$\text{list\_permutation}(L,[X|P]) :- \text{del}(X,L,L1), \text{list\_permutation}(L1,P).$

و بهذا نكون قد انتهينا من البرنامج و في حال سألناه:

?- list\_permutation([a,b,c],X).

فإن النتيجة ستكون:

X = [a,b,c] ? ;

X = [a,c,b] ? ;

X = [b,a,c] ? ;

X = [b,c,a] ? ;

X = [c,a,b] ? ;

X = [c,b,a] ? ;

يمكنك تحميل النص البرمجي من الرابط الخاص به في المراجع.

من أكثر المشاكل التي تواجهني في التعامل مع القوائم الكبيرة هي التكرار، كأن أقوم بإرسال بريد إلكتروني لألفي بريد فأكتشف أن عدد العناوين الحقيقية التي أمتلكها هي 700 و أن 1300 هي عناوين مكررة. في هذا المثال سوف

نقوم بكتابة برنامج بلغة البرولوج لحذف العناصر المكررة في القائمة و كما سبق أن أشرت فنحن نتعامل مع التقنيات ذاتها و إنما بمنطق مختلف.

nodoubles([],[]).

nodoubles([X],[X]).

nodoubles([X1,X2|TAIL],Z):-

list\_delete(X1,[X2|TAIL],Y2),nodoubles(Y2,Z1),list\_append(X1,Z1,Z).

list\_append(A,T,T):-member(A,T).

list\_append(A,TAIL,[A|TAIL]).

list\_delete(\_,[],[]).

list\_delete(Y,[Y],[]).

list\_delete(X,[X|LIST1],LIST1).

list\_delete(X,[Y|LIST],[Y|LIST1]):-list\_delete(X,LIST,LIST1).

كل ما عليك القيام به هو تحميل البرنامج من الرابط المخصص له في المراجع، ثم تشغيله و تجربته كأن تسأله:

?- nodoubles([a,a,b],Result).

ليقوم البرنامج بإظهار a مرة واحدة في القائمة، ثم تعود مرة أخرى لقراءة النص البرمجي و أقسم لك أنك قادر على فهمه بمنهى البساطة. وهذا أكون قد اكتفيت من القوائم بلغة البرولوج و يمكنك العودة دائما لمراجع أخرى أكثر تخصصا و شمولية لمعرفة المزيد، ولكني أعتقد بأن ما قمت بشرحه بالنسبة للقوائم أكثر من كاف لفهم المنطق المتبع في التعامل معها إلا إن كنت تبحث عن نصوص برمجية جاهزة مع قليل من الشرح لتساعدك على تأدية المشاريع المطلوبة منك في الجامعة و في هذه الحالة أقول لك بكل صدر رحب "حل عني - مش فاضيلك" و لكنك ستجد مع هذا المرجع نصوصا برمجية مكتوبة بلغة البرولوج يمكنك تحميلها و تشغيلها فوراً و إن كنت تمتلك ما يكفي من المعلومات حول لغة البرولوج فسوف تكون قادرا على قراءة هذه النصوص و فهمها و تحليلها.

### عوامل المقارنة بلغة البرولوج Comparison Operators

أعتقد أنني قد أشرت إلى بعض هذه العوامل من خلال الأمثلة السابقة في الأجزاء السابقة ولكن الجدول التالي يوضح بالتفصيل هذه العوامل و يشرح وظيفة كل منها.

العامل	وظيفته
>	و هذه الإشارة كما نعرفها جميعا هي إشارة "أكبر من" كأن نقول $X > Y$ أي أن X أكبر من Y

<	و هذه الإشارة كما نعرفها جميعا هي إشارة "أصغر من" كأن نقول $X < Y$ أي أن $X$ أصغر من $Y$
$>=$	أكبر من أو يساوي
$=<$	أصغر من أو يساوي
$=:=$	قيمتين متساويتين كأن نقول قيمة $X$ تساوي قيمة $Y$ $X:=Y$
$=\backslash=$	قيمتين غير متساويتين كأن نقول قيمة $X$ لا تساوي قيمة $Y$ $X=\backslash=Y$

## الخلاصة

بهذا أكون قد انتهيت من سلسلة "تعلم لغة البرولوج" و التي لا أدعي أنها كافية لمن يريد اتقان هذه اللغة ولكني آمل أن أكون قد استطعت خلال هذه السلسلة أن أقدم شيئا للطالب العربي الذي يبحث عن المعلومة بلغته وآمل أن أكون قد ساهمت في إنجاز مادة علمية عربية تستحق القراءة.

خلال الساعات القليلة القادمة يمكنكم إيجاد هذه السلسلة على موقع [www.kutub.info](http://www.kutub.info) ، فقط قم بالبحث عن "مجدي عوض" لتحميل هذه السلسلة مع أمثلة تطبيقية اضافية.

أشكر كل من كان له دور في إتمام هذه السلسلة و الشكر موصول لفريق حملة "أفكار".

تاريخ النشر: 2019/01/07

تأليف: مجدي عوض

## المصادر:

تذكر المراجع بتقنية الـ MLA مع بعض التصرف.

المؤلف	عنوان المصدر	عنوان الموضوع	المساهمون	الإصدار	رقم المجلد	الناشر	تاريخ النشر	تاريخ الاطلاع على المادة	الرابط
Patrick Blackburn, Johan Bos, and Kristina Striegnitz	موقع إلكتروني Learn Prolog !Now	Learn Prolog Now!	Patrick Blackburn, Johan Bos, Kristina Striegnitz	الطبعة الأولى	-	-	2012	2018/12/28	<a href="http://www.learnprolognow.org">/http://www.learnprolognow.org</a>

لتحميل النصوص البرمجية الواردة في هذا البحث قم بالضغط على هذا الرابط:

[www.webtech-internet.com/prolog3.zip](http://www.webtech-internet.com/prolog3.zip)