كلية العلوم

القسم : **الرياضيات**

السنة : **الرابعة**

المادة : ذكاء صنعي

المحاضرة : الرابعة /عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر(What's app-Telegram)على الرقم 0931497960

عملي ذكاء صنعي

الجلسة الرابعة

م.ريم بصل

م.ريم رقبو

تعريف الاستدعاء الذاتي :

هو قابلية البرنامج استدعاء نفسه، أي سوف تعيد الدالة تكرار نفسها (استدعاء نفسها) داخل نفسها.

أنواع الاستدعاء الذاتي:

الاستدعاء الخلفي (نهاية الاستدعاء): تم شرحه بالجلسة السابقة.

الاستدعاء الامامي(رأس الاستدعاء).

النوع الثاني:

هذا النوع مشابه للنوع الأول من كونه يحتوي على شرطي التوقف وشرط الاستدعاء الذاتي ولكنه يختلف بكونه لا يوجد إيعاز بعد عملية الاستدعاء الذاتي لذلك عندما نصل إلى شرط التوقف نحصل على الناتج النهائي.

مثال١:

اكتب برنامج لإيجاد القاسم المشترك الأكبر للعددين صحيحين.

أن القاسم المشترك الأكبر للعددين هو أكبر رقم يقبل علية العددان القسمة بدون باقي. لحل هذا السؤال نحسب باقي قسمة العدد الأول على الثاني إذا كان الناتج صفر فالعدد الثاني هو القاسم المشترك الأكبر وإذا لم يكن باقي القسمة صفر نحسب باقي قسمة العدد الأول على الثاني ونقدم العدد الثاني ليصبح هو الأول وباقي القسمة هو الثاني ونكرر العملية إلى أن يصبح باقي قسمة العدد الأول على الثاني صفر.

مثال رياضي:

gcd(38, 12)=38 mod 12=2
gcd(12, 2)=12 mod 2=0

هنا نتوقف ويكون القاسم المشترك الأكبر: 2

البرنامج:

gcd(M, N, N):- 0 is mod(M,N) ,!.
gcd(M, N, A):- B is mod(M,N) , gcd(N, B, A).

شرط التوقف إذا باقي قسمة الرقم الأول على الثاني تساوي صفر.

لو استدعينا:

gcd(38, 6, X).

الحل:

R1:M=38 N=6 fail.
R2: M=38 N=6 B=2 gcd(6, 2, A)

Call gcd(6, 2, A).

R1:M=6 N=2

هنا عندما نصل إلى شرط التوقف ويتطابق تكون النتيجة هي نفس قيمة Nوهي 2 يكون هو القاسم المشترك الأكبر للعددين 38و6

اكتب برنامج لإيجاد باقي قسمة عددين صحيحين بدون استخدام دالة (mod).

أن أيجاد باقي القسمة للعددين صحيحين يكون بالشكل الآتي:

13 mod 3 = 1

إذ ان حاصل القسمة هو 4وباقي القسمة يكون 1لإيجاد النتيجة بدون استخدام الدالة الجاهزة، إذ نستمر بعملية الطرح إلى أن نصل حد يكون الطرف الأول اصغر من الطرف الثاني هنا يتم التوقف ويكون الرقم المتبقي هو باقي القسمة فمثلاً الرقمان السابقين يكون حسابهما بالشكل الآتي:

13-3=10

10-3=7

7-3=4

4-3=1

نلاحظ ان بعملية الطرح تستمر طالما كان الناتج أكبر من 3، عندما يصل إلى 1وهو أصغر من 3يكون هو باقي القسمة.

البرنامج:-

```
modno(M, N, M):-N>M, !.
modno(M,N,A):-Z is M-N , modno(Z,N,A).
```

لو كان الاستدعاء:

modno(13, 3, X).

الحل:

R1:M=13 N=3 fail.

R2:M=13 N=3 Z=10 modno(10, 3, A)

Call modno(10, 3, A).

R1:M=10 N=3 fail

R2:M=10 n=3 Z=7 modno(7, 3, A)

Call modno(7, 3, A).
R1:M=7 N=3 fail
R2:M=7 N=3 Z=4 modno(4, 3, A)
call modno(4, 3, A).
R1: M=4 N=3 fail
R2: M=4 N=3 Z=1 modno(1, 3, A)
call modno(1, 3, A).
R1: M=1 N=3 match return 1

نلاحظ في هذا عند استدعاء الهدف فأن الحالات الأربعة الأولى في الجدول أن R1 لا تتطابق لكون قيمة M اكبر من قيمة Nلذلك كلها لا تنفذ بينما تنفذ R2 وعندما نصل إلى أخر استدعاء تنفذ R1، إذ تكون قيمة N اكبر من قيمة Mوهنا يتطابق شرط التوقف وينفذ ويكون الرقم الناتج هو Mويكون 1وعندها يتوقف البرنامج.

ايعاز القطع (معامل القطع) cut :

يرمز لمعامل القطع بالرمز (!) ويستخدم للوصول الى اول حل ممكن وقطع (أي حذف) بقية الحلول الممكنة ولاينتقل للقاعدة التالية الا اذا فشلت الأولى ولم تعطي أي حل.

مثال:

a(2). a(5).
b(3). b(2).
c(1). c(2). c(3).

ماهو ناتج الاستفسارات التالية:

1-a(X),!,b(Y),c(Z).

الحل:

X=2 Y=3 Z=1 ,
X=2 Y=3 Z=2 ,
X=2 Y=3 Z=3 ,
X=2 Y=2 Z=1 ,
X=2 Y=2 Z=2 ,
X=2 Y=2 Z=3.

بالتنفيذ:

X = 2,
Y = 3,
Z = 1 ;
X = Z, Z = 2,
Y = 3 ;
X = 2,
Y = Z, Z = 3 ;
X = Y, Y = 2,
Z = 1 ;
X = Y, Y = Z, Z = 2 ;
X = Y, Y = 2,
Z = 3.

2-a(X),b(Y),!,c(Z).

الحل:

X=2 Y=3 Z=1 ,
X=2 Y=3 Z=2 ,
X=2 Y=3 Z=3,

بالتنفيذ:

X = 2,
Y = 3,
Z = 1 ;
X = Z, Z = 2,
Y = 3 ;
X = 2,
Y = Z, Z = 3.

ايعاز النفي fail:

تفشل القاعدة الحالية وننتقل للقاعدة التالية للبحث عن حلول أخرى في حال القاعدة الثانية حوت استدعاء للقاعدة الأولى فستفشل أيضا اما اذا كانت لاتحوي استدعاء فستعطي حلول أخرى.

مثال:

```
p(1). p(2). p(3).
all(X):-write(X),nl,fail.
```

ماهو خرج الاستفسارات التالية:

```
1-p(X),all(X).
```

الحل:

```
1
2
3
false.
```

```
2-p(X).
X=1 ,
X=2 ,
X=3.
```

ايعاز القطع مع ايعاز الفشل (cut & fail):

يفشل القاعدة الحالية ولا ينتقل للقاعدة التالية .

مثال:

```
pet(dog).
pet(cat).
likes(dog,bone).
likes(cat,fish).
likes(cow,grass).

happy_pet(X,Food):-pet(X),likes(X,Food),!.

happy_pet(X,Food):-likes(X,Food).
```

ماهو خرج الاستعلام

```
happy_pet(X,fish).

X=cat.
```

مثال:

```
pet(dog).
pet(cat).
likes(dog,bone).
likes(cat,fish).
likes(cow,grass).

happy_pet(X,Food):-pet(X),likes(X,Food),fail.

happy_pet(X,Food):-likes(X,Food).
```

ماهو خرج الاستعلام:

```
happy_pet(X,Food).

X=dog    Food=bone;
X=cat    Food=fish;
X=cow    Food=grass.
```

مثال:

```
pet(dog).
pet(cat).
likes(dog,bone).
likes(cat,fish).
likes(cow,grass).

happy_pet(X,Food):-pet(X),likes(X,Food),!,fail.

happy_pet(X,Food):-likes(X,Food).
```

ماهو خرج الاستعلام:

```
happy_pet(X,Food).

false.
```

مثال:

```
color(red).
color(green).
color(blue).
first_color(X):-color(X),!.
my_max(X,Y,X):-X>=Y,!.
my_max(_,Y,Y).
print_colors :- color(X),write(X),nl,fail.
print_colors.
not(P):-call(P),!,fail.
not(_).
```

ماهو خرج الاستعلام:

```
1-color(X).
X=red,
X=green,
X=blue.

2- first_color(X).
X=red.

3-my_max(5,3,M).
M=5.
```

```
4-my_max(2,7,M).
M=7.

5-print_colors.
red
green
blue
true.

6-not(color(red)).
false.

7-not(color(pink)).
true.
```

مكتبة A to Z