TP de programmation fonctionnelle et logique
Corrigé du TP 5 : décomposition d'un entier en facteurs premiers

Ce TP est à rendre impérativement pour le mercredi 30 mai, délai de rigueur. Vous devez envoyer votre programme par mail à l'adresse : `vivien@icps.u-strasbg.fr`. Vous pouvez réaliser ce TP seul ou en binôme.

Le but de ce TP est de réaliser la décomposition en facteurs premiers d'un entier. Ainsi, au nombre « 12 », nous voulons que ProLog associe la décomposition : « 2 * 2 * 3 ».

Dans tout le sujet nous supposerons que nous ne manipulons que des **entiers strictement positifs**.

1. Un nombre est premier s'il n'est divisible que par 1 et par lui-même. Écrivez un prédicat `nonpremier` qui est vrai si et seulement si le nombre qui lui ait passé en argument n'est pas premier, c'est-à-dire si et seulement si ce nombre admet un diviseur non trivial, ou est égal à un.

   ```
   nonpremier(1).
   nonpremier(X) :- Y is X-1, between(2,Y,Z), between(Z,Y,T), X =:= T*Z.
   ```

   Exemples d'utilisation :

   ```
   ?- nonpremier(1).
   Yes
   ?- nonpremier(7).
   No
   ?- nonpremier(12).
   Yes
   ```

2. Écrivez, à partir du prédicat `nonpremier`, un prédicat `premier` qui est vrai si et seulement si le nombre qu'il prend en argument est premier.

   ```
   premier(X) :- nonpremier(X), !, fail.
   premier(_).
   ```

   Exemples d'utilisation :

   ```
   ?- premier(1).
   No
   ?- premier(7).
   Yes
   ?- premier(12).
   No
   ```

3. Écrivez un prédicat `touslespremiers`, prenant en argument deux nombres `X` et `Y`, et vrai si `X` est un nombre premier inférieur ou égal à `Y`. Ce prédicat doit pouvoir énumérer tous les nombres premiers `X` inférieurs ou égaux à une valeur `Y` donnée.

   ```
   touslespremiers(X,Y):- between(2,Y,X), premier(X).
   ```

   Exemple d'utilisation :

   ```
   ?- touslespremiers(X,9).
   X = 2 ;
   X = 3 ;
   X = 5 ;
   ```

```
X = 7 ;
No
```

4. Écrivez, sans utiliser la primitive de division, un prédicat `division(A,B,Q,R)` où `Q` et `R` sont respectivement le quotient et le reste de la division de `A` par `B` (pour l'utilisation du prédicat, on suppose que `A` et `B` sont donnés, et que l'on recherche `Q` et `R`).

```
division(A,B,Q,R) :-
Bmoins1 is B-1, between(0,Bmoins1,R), between(0,A,Q), A =:= B*Q+R.
```

autre solution :

```
division(A,B,0,A):- A<B.
division(A,B,Q,R):- A >= B, BmoinsA is A - B,
                    division(BmoinsA, B, Qmoins1, R),
                    Q is Qmoins1 + 1.
```

5. Écrivez, à partir du prédicat `division`, un prédicat `divise` prenant en entrée trois entiers `A`, `B` et `C` et qui est vrai si `A` est divisible par `B` de quotient `C`. Ce prédicat devra être capable de calculer `C` connaissant `A` et `B`.

```
divise(A,B,C):-division(A,B,C,0).
```

Exemple d'utilisation :

```
?- divise(12,2,C).
C = 6
Yes
```

6. Écrivez une fonction `decomposition` prenant en entrée deux variables `X` et `D` et qui, à `X` entier de valeur donnée, associe `D`, une de ses décompositions en facteurs premiers.

```
decomposition(X,X):-premier(X),!.
decomposition(X,D):-touslespremiers(Y,X), divise(X,Y,Z),
                    decomposition(Z,E), D = Y * E.
```

Exemples d'utilisation :

```
?- decomposition(7,X).
X = 7 ;
No
?- decomposition(12,X).
X = 2* (2*3) ;
X = 2* (3*2) ;
X = 3* (2*2) ;
No
```

Note : les parenthèses n'ont ici aucune espèce d'importance...

7. Modifiez le prédicat précédent pour qu'il ne produise plus qu'une décomposition en facteurs premiers, celle dans laquelle les facteurs sont rangés dans l'ordre croissant.

```
decompositionbis(X,X):-premier(X),!.
decompositionbis(X,D):-touslespremiers(Y,X), divise(X,Y,Z), !,
                       decompositionbis(Z,E), D = Y * E.
```

Exemples d'utilisation :

```
?- decompositionbis(12,X).
X = 2* (2*3) ;
```

```
No
?- decompositionbis(2001,X).
X = 3* (23*29) ;
No
```

8. Écrivez un prédicat `pluspetitpremier` prenant en entrée deux variables `X` et `Y` et qui est vrai si et seulement si `X` est le plus petit entier premier supérieur ou égal à l'entier `Y` de valeur donnée.

```
pluspetitpremier(X,X):- premier(X), !.
pluspetitpremier(X,Z):- Y is Z+1, pluspetitpremier(X,Y).
```

Exemples d'utilisation :

```
?- pluspetitpremier(X,97).
X = 97 ;
No
?- pluspetitpremier(X,98).
X = 101 ;
No
```