

COMMENT FAIRE ÉVOLUER UN EXERCICE EN UN EXERCICE WIMS

BERNADETTE PERRIN-RIOU

7 MAI 2003

1. LA STRUCTURE D'UN EXERCICE

Donnons d'abord quelques indications sur la structure d'un exercice.

1.1. Informations préliminaires. On commence par des informations préliminaires

```
\title{Essai}
\language{fr}
\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{yes}
\format{html}
\precision{10000}
```

`range` donne les bornes qui permettent de tester si les réponses sont correctes lorsqu'il s'agit de fonctions

`computeanswer` dit si la réponse est calculée avant d'être comparée au résultat. Par exemple, si l'argument de `computeanswer` est `no`, $\cos(0)$ ne sera pas accepté à la place de 1.

`precision` concerne la précision numérique demandée dans les réponses (erreur relative).

1.2. Définition des objets. Ensuite arrivent toutes les commandes définissant les objets dont il sera question dans l'énoncé. Cela peut être des données mathématiques, des données concernant le texte de l'énoncé. Une fois défini par une commande du type `a=...`, l'objet `a` correspondant peut être réutilisé sous la forme `\a`

Les types pouvant être utilisés sont les suivants

```
\text{} \real{} \integer{} \rational{} \complex{} \function{}
```

A l'intérieur, on peut ensuite mettre un certain nombre d'instructions. Citons parmi les plus utiles pour le début `randint`, `random`, `randitem`, `item`, des commandes mathématiques simples `diff`, `int`, `simplify`. Par exemple :

```
\text{choix= oui, non}
\integer{n=randint(2..100)} on prend un entier aléatoire entre 2 et 100
\integer{m=randint(4..10)} on prend un entier aléatoire entre 4 et 10
\rational{a=\n/\m} on prend le rationnel n/m
\real{b=(\a)^2}
\integer{c=\b}
\text{M=random(0,1), random(0,1,2)}
\text{m1=item(1,\M)}
\text{m2=item(2,\M)}
\function{f=simplify(x+ (\a)*x^m1+(\c)*m2)}
\real{freel=evaluate(\f,x=\a)}
\rational{frat= evaluate(\f,x=\a)}
```

```
\text{choix=oui,non}
\text{reponse=\n%2=0? item(1,\choix): item(2,\choix)}
```

Pour voir le résultat de ces commandes, on peut mettre les variables définies dans la partie énoncé.

1.3. **L'énoncé.** La question est ensuite mise dans la commande `statement`. Ce qui suit est ce qui est rentré, pas ce que l'on voit à l'écran, et bien sûr dans un vrai exercice, on ne mettra pas les réponses et tous les nombres calculés !

```
\statement{Voici les nombres qui ont été calculés dans
la première partie :
n=\n, m=\m, a=\a, b=\b, c=\c, M=\M, m1=\m1, m2=\m2,f=\f, freel=\freel, frat=\frat
<p>En particulier, que vaut \f en \ (x=\a) d'abord comme un rationnel, puis comme un r
Etes vous d'accord que \n est pair ?
```

Il y a ici trois questions, il faudra donc trois réponses. les deux premières sont de type "réponses libres", la troisième est de type "choix multiples".

On peut remarquer aussi qu'on a utilisé une commande `html` pour aller à la ligne. D'autre part, si vous essayez ce texte, la fonction f apparaît avec les `*`. Si vous voulez une forme plus jolie, il vaut mieux écrire

```
\statement{Voici les nombres qui ont été calculés dans
la première partie :
n=\n, m=\m, a=\a, b=\b, c=\c, M=\M, m1=\m1, m2=\m2,f=\f, freel=\freel, frat=\frat
<p>En particulier, que vaut \(\f) en \ (x=\a) d'abord comme un rationnel,
puis comme un réel.}
```

Ainsi, le résultat de f sera réinterprété par WIMS directement pour le transformer en un version "imprimable" et non en syntaxe ordinateur, puis mis à l'écran soit en `html`, soit en `tex` selon la complexité de la formule. De manière générale, les formules doivent être encadrées par

```
\( ... ) ou \ ( ... \)
```

1.4. **La ou les réponses.** Les réponses se trouvent dans les champs `answer` ou `choice` selon qu'il s'agit d'une réponse ou d'un choix parmi des choix proposés. Par exemple,

```
\answer{En réel}\freel{type=numeric}
\answer{En rationnel}\frat{type=litexp}
\choice{Parité de \n}\reponse{\choix}
```

On voit apparaître en troisième argument le type de la réponse.

1.5. **Champs supplémentaires.** D'autres champs peuvent apparaître : `help`, `hint`, `solution`, `feedback`. Nous ne les détaillons pas ici.

2. COMMENT FAIRE ÉVOLUER UN EXERCICE SIMPLE

Nous allons prendre un exercice simple et le faire évoluer en introduisant quelques potentialités de WIMS. Dans le texte, nous ferons apparaître les évolutions de l'exercice sans reprendre tout à chaque fois. A la fin, se trouvent les versions complètes des exercices.

Prenons l'exercice simple suivant :

Trouver la décomposition en binaire du nombre 123.

2.1. **Première évolution très simple : remplacer 123 par un entier quelconque n .** Il va falloir faire un choix dans la conception de l'exercice : vaut-il mieux choisir au hasard n et calculer dans la partie "calculs" sa décomposition en binaire : Ou bien, va-t-on choisir une suite aléatoire de k entiers égaux à 0 et de 1, calculer l'entier correspondant n et fournir n dans la question. Nous allons choisir la seconde méthode en choisissant pour l'instant la longueur k de la suite petite :

```
\integer{k=4}
\integer{n1=random(1,0)}
\integer{n2=random(1,0)}
\integer{n3=random(1,0)}
\integer{n4=random(1,0)}
\integer{n= \n1+\n2*2+\n3*2^2+\n4*2^3}
\integer{nbin= \n1+\n2*10+\n3*10^2+\n4*10^3}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{numeric}
```

Si l'on veut choisir un k plus grand, on pourra éprouver le besoin de faire une boucle très simple

```
\integer{k=7}
\integer{n=0}
\integer{nbin=0}
\for{i=1 to \k}{
\integer{n1=random(1,0)}
\integer{n= \n+\n1*2^(\i-1)}
\integer{nbin= \nbin+\n1*10^(\i-1)}
}
\text{nbin=pari(\nbin)}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{litexp}
```

Essayer. Le but de la ligne

```
\text{nbin=pari(\nbin)}
```

est que la réponse apparaisse comme un entier et non en notation scientifique. On fait pour cela appel à un logiciel externe Pari/GP. Si on voulait augmenter l'entier k , on aurait d'ailleurs intérêt à faire tout le calcul dans Pari/GP. Mais cela fait partie de possibilités avancées : par exemple :

```
\text{S=pari(k=10;n=0;nbin=0; for(i=1,k, n1=RANDOM(2);
n=n+n1*2^(i-1);
nbin=nbin+n1*10^(i-1)); print(n","nbin))}
\text{n=item(1,\S)}
\text{nbin=item(2,\S)}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{litexp}
```

Ici, ce qui est à l'intérieur de la parenthèse `pari(. .)` est un programme écrit en GP/Pari. On aurait pu aussi écrire un programme (de quelques lignes) en utilisant d'autres logiciels, pour l'instant maxima, yacas.

2.2. **Feedack.** Dans l'exercice précédent, une faute grave de l'élève serait de faire apparaître dans sa réponse un chiffre qui ne serait pas 0 ou 1. Il est possible de le reprendre si cela arrive en ajoutant un feedback de la manière suivante. Après les champs `answer`, on peut faire des opérations sur les réponses, les analyser et mettre ensuite un avertissement. On fait alors appel aux réponses à l'aide de

reply i ou

choice i où i est le numéro de la question ou réponse (du type réponse libre ou choix).

On rajoute donc

```
\text{test=0}
\for{i=2 to 9}{
\if{\i isin \reply1}{\text{test=1}}
}
\feedback{\test=1}{Attention, dans la décomposition binaire d'un entier,
seuls les chiffres 0 ou 1 apparaissent.}
```

2.3. Deuxième évolution. On voudrait maintenant guider l'élève et pas seulement lui demander le résultat. Il y a pour cela plusieurs manières qui dépendent de la manière dont la matière sur laquelle porte l'exercice est enseignée.

Par exemple, on peut demander à l'élève de remplir un tableau en faisant apparaître les valeurs explicites des puissances de 2 comme le tableau suivant :

Puissances de 2	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	64	32	16	8	4	2	1
34 = ? + ? + ?							

Pour transformer cette méthode, plusieurs variantes sont certainement possibles (on ne parle pas encore de la technique) pouvant correspondre à un niveau de compréhension ou d'apprentissage.

- (1) Ecrire explicitement la décomposition à la places des ? ou la faire écrire ?
- (2) Fournir quelques-uns des chiffres de la réponse ?
- (3) Fournir les premiers chiffres (sur la gauche)
- (4) N'en fournir aucun
- (5) Fournir les valeurs des puissances de 2 ou quelques-unes ou aucune ?

Tout dépend de ce qu'on attend des élèves, de leur niveau. Ainsi, il peut être intéressant de faire plusieurs de ces variations (le schéma de programmation est quasiment identique) et de l'utiliser successivement dans sa classe.

2.4. Techniques. On a vu apparaitre un tableau, on a envie de faire remplir des cases. Pour transformer cela en un exercice interactif, nous allons utiliser

- la possibilité de rentrer des commandes html
- la commande `embed` qui permet de faire apparaître des champs de réponses.

Nous allons aussi avoir besoin de la liste des 0 et des 1. Pour cela, on reprend la première partie en l'adaptant afin de conserver la liste de 0 et de 1

```
\integer{k=3}
\integer{n=0}
\text{liste=2}
\for{i=1 to \k}{
\integer{n1=random(1,0)}
\integer{n= \n+\n1*2^{(i-1)}}
\text{liste= \liste, \n1}
}
\text{nbin=pari(\nbin)}
\text{rep1=item(\k+1,\liste)}
\text{rep2=item(\k,\liste)}
\text{rep3=item(\k-1,\liste)}
```

Nous ne faisons ici que la version où l'on demande toutes les réponses dans la troisième ligne. Pour faire ce tableau en html, les commandes html (version simple) sont

```
<table align="center" border=1>
  <tr>
    <td>Puissances de 2</td>
    <td> 2<sup>2</sup></td>
    <td>2<sup>1 </sup></td>
    <td>2<sup>0 </sup>
  </td>
</tr>
<tr>
<td>&nbsp;</td>
  <td>4</td>
  <td>2</td>
  <td>1</td>
</tr>
<tr>
<td> 5</td>
<td>..</td>
<td>...</td>
<td>...</td>
</tr>
</table>
```

Si l'on veut maintenant remplacer les ... par un champ de réponses :on met dans la partie `statement` :

```
<table align="center" border=1>
  <tr>
    <td>Puissances de 2</td>
    <td> \ (2^2)</td>
    <td>\ (2^1)</td>
    <td>\ (2^0) </td>
  </tr>
<tr>
<td>&nbsp;</td>
  <td>4</td>
  <td>2</td>
  <td>1</td>
</tr>
<tr>
  <td> \n</td>
  <td>\embed{reply 1,4}</td>
  <td>\embed{reply 2,4}</td>
  <td>\embed{reply 3,4}</td>
</tr>
</table>
```

Le deuxième argument de `embed` est la taille de la case de formulaire.

Mais il faut bien sûr changer les réponses

```
\answer{}{\rep1}{type=numeric}
\answer{}{\rep2}{type=numeric}
```

```
\answer{}{\rep3}{type=numeric}
```

On pourrait aussi construire le tableau en le programmant. Mais nous serons quand même limiter par le fait qu'on ne peut pas construire les `embedreply i` à l'aide d'une boucle.

Mais pour ceux qui accrochent! c'est assez instructif. On pourrait par exemple imaginer cela dans un document où la décomposition binaire d'un entier serait calculée avec les étapes, celles ci variant bien sûr selon les entiers.

On va donc construire le tableau. On peut le faire dans la partie programme ou directement dans l'énoncé. Dans la partie programme : Nous allons construire à la fois la ligne des puissances et la ligne des puissances effectuées

```
\text{puissance= <tr><td>Puissances de 2</td>}
\text{puisscalcul= <tr><td></td>}
\for{i=1 to \k}{
\text{puissance=\puissance
<td>2<sup>\i</sup> </td>}
\integer{deuxi=2^\i}
\text{puisscalcul=\puisscalcul
<td>\deuxi </td>}
}
\text{puissance=\puissance
</tr>}
\text{puisscalcul=\puisscalcul
</tr>}
```

Dans la partie énoncé, les deux premières lignes seront obtenues de la manière suivante

```
<table align="center">
\puissance
\puisscalcul
</table>
```

Mettons la dernière ligne du tableau directement dans le statement. Et là j'ai peur qu'il ne faille le faire à la main! (heureusement qu'il y a le copier-coller)

```
<table><tr>
\for{i=1 to \k}{<td>\embed{reply \i, 3} </td>
</tr></table>
```

On fixe $k = 6$, cela suffit bien

```
<table align="center">
\puissance
\puisscalcul
<tr><td>\n=</td><td> \embed{reply 1,2}</td>
<td> \embed{reply 2,2}</td><td>\embed{reply 3,2}</td>
<td> \embed{reply 4,2}</td><td> \embed{reply 5,2}</td>
<td> \embed{reply 6,2}</td><td> \embed{reply 7,2}</td></tr>
</table>
```

On peut aussi donner la réponse pour certains et pas pour d'autres.

Si on regarde le résultat, on peut avoir envie que la réponse soit écrite sous la forme traditionnelle. On peut alors le faire et faire écrire tout ce qu'on veut dans un des champs `solution` ou `feedback`

```
\feedback{1=1}{L'écriture de \n en binaire est \nbin}
```

2.5. **Troisième évolution.** Pour faire apparaître une des réponses (mais pas toujours au même endroit), on peut utiliser `if` soit dans le texte de la question, soit dans la partie des définitions, soit dans les deux.

ANNEXE A. LES DIFFÉRENTES VERSIONS

A.1. Version 1.

```
\title{Binaire1}
\language{fr}
\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{no}
\format{html}
\precision{10000}

\integer{k=4}
\integer{n1=random(1,0)}
\integer{n2=random(1,0)}
\integer{n3=random(1,0)}
\integer{n4=random(1,0)}
\integer{n= \n1+\n2*2+\n3*2^2+\n4*2^3}
\integer{nbin= \n1+\n2*10+\n3*10^2+\n4*10^3}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{numeric}
```

A.2. Version 2.

```
\title{Binaire2}
\language{fr}
\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{no}
\format{html}
\precision{10000}

\integer{k=7}
\integer{n=0}
\integer{nbin=0}
\for{i=1 to \k}{
\integer{n1=random(1,0)}
\integer{n= \n+\n1*2^(\i-1)}
\integer{nbin= \nbin+\n1*10^(\i-1)}
}
\text{nbin=pari(\nbin)}
\statement{\nbin Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{litexp}
```

A.3. Version 3.

```
\title{Binaire3}
\language{fr}
```

```

\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{no}
\format{html}
\precision{10000}
\text{S=pari(k=10;n=0;nbin=0; for(i=1,k,
n1=RANDOM(2); n=n+n1*2^(i-1);
nbin=nbin+n1*10^(i-1)); print(n","nbin))}
\text{n=item(1,\S)}
\text{nbin=item(2,\S)}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{litexp}

```

A.4. Version 4.

```

\title{Binaire4}
\language{fr}
\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{no}
\format{html}
\precision{10000}

\integer{k=7}
\integer{n=0}
\integer{nbin=0}
\for{i=1 to \k}{
\integer{n1=random(1,0)}
\integer{n= \n+\n1*2^(\i-1)}
\integer{nbin= \nbin+\n1*10^(\i-1)}
}
\text{nbin=pari(\nbin)}
\statement{Ecrire le nombre \n en binaire.}
\answer{en binaire, \n vaut}{\nbin}{litexp}

```

```

\text{test=0}
\for{i=2 to 9}{
\if{\i isin \reply1}{\text{test=1}}
}

```

\feedback{\test=1}{Attention, dans la décomposition binaire d'un entier, seuls les chiffres 0 ou 1 apparaissent.}

A.5. Version 5.

```

\title{Binaire5}
\language{fr}
\range{-5..5}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\computeanswer{no}

```

```

\format{html}
\precision{10000}
\integer{k=3}
\integer{n=0}
\text{liste=2}
\for{i=1 to \k}{
\integer{n1=random(1,0)}
\integer{n= \n+\n1*2^(\i-1)}
\text{liste= \liste, \n1}
}
\text{nbin=pari(\nbin)}
\text{rep1=item(\k+1,\liste)}
\text{rep2=item(\k,\liste)}
\text{rep3=item(\k-1,\liste)}

\statement{ Compléter le tableau pour écrire le nombre \n en binaire.

<table align="center" border=1>
  <tr>
    <td>Puissances de 2</td>
    <td> \ (2^2)</td>
    <td>\ (2^1)</td>
    <td>\ (2^0) </td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>4</td>
    <td>2</td>
    <td>1</td>
  </tr>
  <tr>
    <td> \n</td>
    <td>\embed{reply 1,4}</td>
    <td>\embed{reply 2,4}</td>
    <td>\embed{reply 3,4}</td>
  </tr>
</table>
}
\answer{}{\rep1}{type=numeric}
\answer{}{\rep2}{type=numeric}
\answer{}{\rep3}{type=numeric}

```