Corrigé TP8

Exercice 08.1

On a $2^n = \underset{n \to +\infty}{o}(n!)$, donc on sait que $\lim\limits_{n \to +\infty} \dfrac{n!}{2^n} = +\infty$.

```
PROGRAM factorielle_puissance ;
        VAR A,u : REAL ;
            n : INTEGER ;
BEGIN
WRITELN('Donner le réel A > 0') ;
READLN(A) ;
n := 1 ;
u := 1/2 ;
WHILE u <= A
    DO
        BEGIN
        n := n+1 ;
        u := u*n/2 ;
        END ;
WRITELN('L''entier n cherché vaut : ',n) ;
READLN ;
END.
```

Exercice 08.2

```
PROGRAM urne ;
        VAR a,n,sauvegarde : INTEGER ;
BEGIN
n := 0 ;
a := random(10)+1 ;
WRITELN('Tirage 1 : ',a) ;
REPEAT
    BEGIN
    sauvegarde := a ;
    a := random(10)+1 ;
    n := n+1 ;
    WRITELN('Tirage ',n,' : ',a) ;
    END
UNTIL sauvegarde = a ;
WRITELN('Première égalité consécutive aux numéros ',n-1,' et ', n) ;
READLN ;
END.
```

Exercice 08.3

Déjà fait dans le TP7.

Exercice 08.4

1. Notons $\forall n \in \mathbb{N}, v_n = u_{2n}$ et $w_n = u_{2n+1}$. On a

$$\forall n \in \mathbb{N}, v_{n+1} - v_n = u_{2n+2} - u_{2n} = \frac{(-1)^{2n+2}}{(2n+2)!} - \frac{(-1)^{2n+1}}{(2n+1)!} = \frac{1}{(2n+2)!} + \frac{1}{(2n+1)!} \geq 0$$

Donc (v_n) est croissante.

$$\forall n \in \mathbb{N}, w_{n+1} - w_n = u_{2n+3} - u_{2n+1} = \frac{(-1)^{2n+3}}{(2n+3)!} - \frac{(-1)^{2n+2}}{(2n+2)!} = -\frac{1}{(2n+3)!} - \frac{1}{(2n+2)!} \leq 0$$

Donc (w_n) est décroissante.

$$\forall n \in \mathbb{N}, v_n - w_n = \frac{(-1)^{2n}}{(2n)!} - \frac{(-1)^{2n+1}}{(2n+1)!} \underset{n \to +\infty}{\longrightarrow} 0$$

Donc les suites (u_{2n}) et (u_{2n+1}) sont adjacentes. Par le théorèmes des suites adjacentes, elles convergent vers une même limite ℓ. On en déduit que la suite (u_n) converge aussi vers ℓ d'après le théorème des suites extraites d'ordre pair et impair.

2. Déjà, remarquons que Supposons que $n = 2p$. Alors $u_n = u_{2p} = v_p$. Donc $|\ell - u_n| = |\ell - v_p| = v_p - \ell$. Mais $\ell \geq w_p$ donc $-\ell \leq -w_p$, donc $v_p - \ell \leq v_p - w_p$. Or,

$$v_p - w_p = \sum_{k=0}^{2p} \frac{(1)^k}{k!} - \sum_{k=0}^{2p+1} \frac{(1)^k}{k!} = \frac{1}{(2p+1)!}$$

On a donc

$$|\ell - u_n| \leq \frac{1}{(n+1)!}$$

Supposons que $n = (2p+1)$. Alors $u_n = u_{2p+1} = w_p$. Donc $|\ell - u_n| = |\ell - w_p| = \ell - w_p \leq v_p - w_p$. Or, on a déjà calculé $v_p - w_p$. et donc

$$|\ell - u_n| \leq \frac{1}{(2p+1)!} = \frac{1}{n!} < \frac{1}{(n+1)!}$$

En conclusion, pour tout $n \in \mathbb{N}$, on a bien

$$|\ell - u_n| \leq \frac{1}{(n+1)!}$$

3. Soit $\varepsilon > 0$. Comme $\frac{1}{(n+1)!} \underset{n \to +\infty}{\longrightarrow} 0$, il existe un rang N tel que $\forall n \geq N, \frac{1}{(n+1)!} < \varepsilon$.
En particulier, on est sûr qu'il existe un rang n tel que $|\ell - u_n| < \varepsilon$ (tout $n \geq N$ convient), c'est-à-dire que u_n sera une valeur approchée de ℓ à ε près.

```
PROGRAM valeur_approchee_while ;
    VAR n,signe :INTEGER ; (* signe va contenir (-1)^n*)
        u,quotient :REAL ; (* quotient va contenir 1/n!*)
BEGIN
n :=0 ;
u :=1 ;
quotient :=1 ;
signe :=1 ;
WHILE quotient>=0.001 DO
    BEGIN
    n:=n+1 ;
    quotient :=quotient/n ;
    signe := signe*(-1) ;
    u := u+signe*quotient ;
    END ;
WRITELN('Une valeur approchée de la limite à 10^-3 près est ',u) ;
READLN ;
END.
```

4. Il suffit de modifier un tout petit peu le programme précédent. Il faut rajouter un message pour que l'utilisateur choisisse la valeur de p et calculer $10^{-p} = \left(\frac{1}{10}\right)^p$: on sait faire, c'est une boucle `FOR`.

```
PROGRAM valeur_approchee_while2 ;
    VAR n,signe,p,k :INTEGER ;
        u,quotient,puiss :REAL ; (* puiss va contenir 10^-p *)
BEGIN
n :=0 ;
u :=1 ;
quotient :=1 ;
signe :=1 ;
puiss :=1 ;
WRITELN('Donner la valeur de l''entier p>0') ;
READLN(p) ;
FOR k :=1 TO p DO puiss :=puiss/10 ;
WHILE quotient>=puiss DO
    BEGIN
    n:=n+1 ;
    quotient :=quotient/n ;
    signe := signe*(-1) ;
    u := u+signe*quotient ;
    END ;
WRITELN('Une valeur approchée de la limite à 10^-',p,' près est ',u) ;
READLN ;
END.
```