

*Durée : 60'**Calculatrice autorisée*

Question 1

14 (8+4+2) points

- (1) Ecrire la fonction **power** qui prend en entrée un réel x de type `extended` et un entier naturel n de type `integer` et qui calcule de façon *aussi rapide que possible* n^{e} puissance de x .
- (2) Donner un tableau d'exécution montrant l'évolution des *paramètres* et *variables* de cette fonction pour l'appel **power(-2,10)**.
- (3) Quelle instruction utilisant la fonction **power** permet d'afficher à l'écran le résultat évalué de l'expression :

$$(a^b - 1)^{a+b},$$

où l'on suppose que a et b sont deux variables bien déclarées et bien définies au préalable. Quel doit être le type de ces variables pour qu'il n'y ait pas de message d'erreur lors de l'exécution de cette instruction ?

Question 2

18 points

Ecrire une procédure **lancer(n:integer)**, qui écrit à l'écran **n** résultats (aléatoires) du lancer d'un dé. Chaque ligne de l'écran de sortie comportera au plus 10 lancers, précédés chacun de 3 blancs (espaces). Ensuite la procédure écrira à l'écran, pour chaque face, le nombre d'occurrences (apparitions) de cette face. L'appel **lancer(45)**, donnera par exemple l'écran de sortie suivant :

```

Voici les resultats de 45 lancers :

  4  2  2  2  5  1  4  2  6  4
  3  3  3  3  1  1  5  4  6  6
  1  5  4  6  4  5  3  6  1  3
  1  2  2  3  5  1  1  2  6  3
  2  4  5  6  1

Face :                1  2  3  4  5  6
-----
Nombre d'occurrences :  9  8  8  7  6  7

```

N.B. : Votre procédure devra fidèlement reproduire l'écran de sortie ci-dessus. On ne demande pas le programme principal, mais uniquement la procédure ! On n'inclura pas l'instruction `randomize` dans la procédure.

Question 3

12 points

Ecrire une procédure **factorisation(n :integer)** qui écrit à l'écran la décomposition en facteurs premiers de l'entier naturel non nul **n**. *Exemple* : Pour **n = 360**, la procédure affichera : **360 = 2*2*2*3*3*5**.

Question 4

16 points

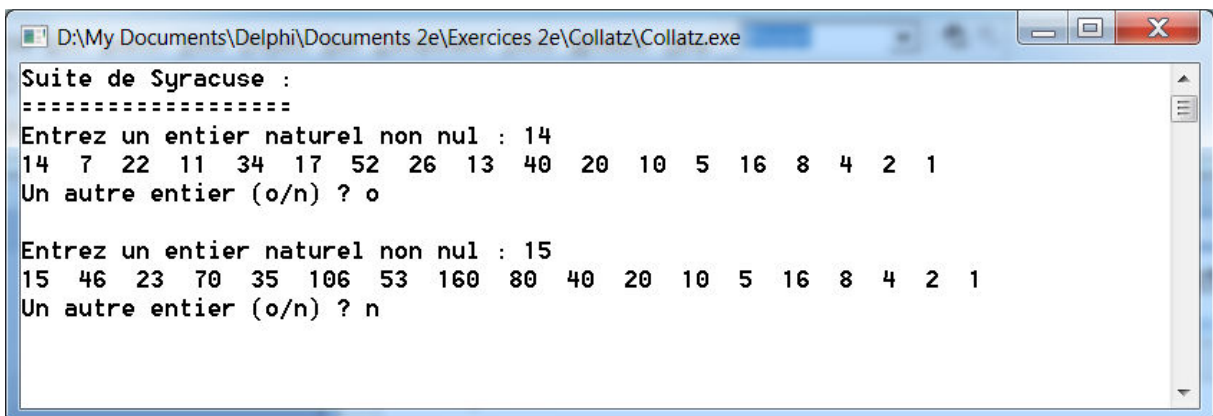
Ecrire un programme qui affiche la *suite de Syracuse* pour des entiers naturels non nuls entrés par l'utilisateur. Cette suite est définie de la manière suivante :

On part d'un entier naturel non nul **n** ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite récurrente appelée suite de Syracuse de l'entier **n**. Par exemple, à partir de 14, on construit la suite des nombres: 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. La suite s'arrête si on tombe sur 1. (On peut supposer que cela est toujours le cas, même si personne ne l'a démontré jusqu'à ce jour ...)

On écrira obligatoirement la fonction **syra** qui est définie de la manière suivante pour un entier naturel non nul :

$$\mathbf{syra(n)} = \begin{cases} n / 2 & \text{si } n \text{ est pair} \\ 3n + 1 & \text{si } n \text{ est impair} \end{cases}$$

Voici la fenêtre d'exécution (*à respecter !*) obtenue lorsqu'on demande la suite de Syracuse des entiers 14 et 15. Deux éléments successifs de la suite seront toujours séparés par deux blancs.



```
Suite de Syracuse :
=====
Entrez un entier naturel non nul : 14
14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Un autre entier (o/n) ? o

Entrez un entier naturel non nul : 15
15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
Un autre entier (o/n) ? n
```

G. Lorang