Série d'exercices n° 5 : Les Algorithmes Arithmétiques

Ecrire un algorithme qui permet de calculer et d'afficher une approximation e jusqu'au rang n en utilisant la série suivante :

$$e = \sum_{i=1}^{\infty} \frac{1}{i!}$$

En utilisant la même série définie ci-dessus, écrire un algorithme qui calcul et affiche une approximation e. On arrête les calculs lorsque la différence entre deux termes consécutifs sera inférieure à 10^{-6}.

Exercice N° 1

Ecrire un algorithme qui lit un nombre réel X, calculer et afficher e^x.

$$e^x = \sum_{i=0}^{n} \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

On arrête les calculs lorsque $\left|\frac{x^i}{i!}\right| \prec 10^{-6}$

Exercice N° 2

L'intégration de $e^{-x^2/2}$ fournit:

$$IE= x + (-x^2/2)/1!*x/3 + (-x^2/2)^2/2!*x/5 + (-x^2/2)^3/3!*x/7 + (-x^2/2)^4/4!*x/9 + (-x^2/2)(-x^2/2)^5/5!*x/11 + ...$$

On arrête les calculs lorsque la différence entre deux termes consécutifs sera inférieure à 10^{-6}.

Exercice N° 3

Cosinus en radian est donnée par la somme infinie suivante :

$$Cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + ...$$

On décide d'arrêter la somme lorsque la différence entre deux terme consécutifs $<10^{-6}$.

Ecrire un algorithme qui permet d'évaluer et d'afficher le cosinus d'une valeur X donnée.

Exercice N° 4

Etablir un algorithme qui calcul et affiche le sinus et le rang du dernier terme i ; sachant que le calcul s'arrête lorsque la différence de deux termes consécutifs sera inférieure à 10^{-7}.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - ......$$

Exercice N° 5

Ecrire un algorithme qui a pour donnée un réel X tel que |X| < 1 et donne pour résultat une valeur approchée (avec une précision = 10^{-6}) de Arctg(X) calculée en utilisant le développement en série :

$$Arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} + ... + (-1)^n x^{2n+1}/(2n+1) + ...$$

On arrête le calcul quand la différence entre deux mono termes de la somme est inférieure à 10^{-6}.

Exercice N° 6

En utilisant l'exercice n° 4, écrire un algorithme permettant de transformer les coordonnées cartésiennes (x,y) en coordonnées polaires (r,t) qui se fait par les formules :

$r^2 = x^2+y^2$

t = arctg (y/x) auquel il faut ajouter pi si x < 0

sauf si x = 0, t = pi/2 si y > 0

t = - pi/2 si y < 0

t n'existe pas si y = 0

Exercice N° 7

Ecrire un algorithme qui détermine et affiche une valeur approchée de Π par la méthode de Wallis, définie par la formule suivante :

$$\frac{\Pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \frac{8}{9} \times .....$$

On arrête le calcul quand la différence entre deux valeurs consécutives de cette formule devient inférieure à 10^{-6}.

Exercice N° 7

Ecrire un algorithme qui calcul la valeur approchée de π sur base de la série suivante. Les calculs s'arrêtent lorsque la valeur du dernier terme ajouté est inférieure à un ε donné.

$$\boxed{\frac{\pi}{4} = 4 \cdot arctan\left(\frac{1}{5}\right) - arctan\left(\frac{1}{239}\right)}$$

Avec la série :

$$arctan(x) = \frac{x}{1+x^2} \sum_{i=0}^{\infty} c_i \left(\frac{x^2}{1+x^2}\right)^i$$

où $c_0 = 1$

et pour $i > 0, c_i = \frac{2 \cdot i}{2 \cdot i + 1} c_{i-1}$

Exercice N° 7

Sachant que **$6+6/2^2 + 6/3^2 + 6/4^2 + \ldots + 6/n^2$ tend vers π^2**, écrire un programme permettant de calculer puis d'afficher une valeur approchée de π^2 avec une erreur maximale ε. La valeur de ε est une donnée.

Ecrire un algorithme qui calcul la valeur approchée de π sur base de la série suivante (due à Leibniz). Les calculs s'arrêtent lorsque la valeur du dernier terme ajouté est inférieure à un ε donné.

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots$$

Ecrire un algorithme qui calcul la valeur approchée de π sur base de la série suivante (due à Leibniz). Les calculs s'arrêtent lorsque la valeur du dernier terme ajouté est inférieure à un ε donné.

$$\frac{\pi}{8} = \frac{1}{1 \cdot 3} + \frac{1}{5 \cdot 7} + \frac{1}{9 \cdot 11} + \cdots$$

Exercice N° 8

On désire remplir une matrice carrée A de dimension N avec 1<=N<=100 d'après la formule suivante :

A[i,j] = PPCM(i,j)

- Ecrire une fonction PGCD qui retourne le pgcd de deux entiers a et b par l'algorithme de décrit ci-dessous :

  1. Déterminer la plus grande puissance K de 2 qui divise A et B à la fois ; remplacer A par A div 2^K et B par B div 2^K
  2. A présent, A ou B est impair. Si A ≠ B, faire ce qui suit :

     T ← |A – B|

     Si T est pair, remplacer T par T div 2. Répéter ceci tant que T est pair.

     A ← T si A > B, B ← T si non.

     Si A ≠ B répéter l'étape 2.
  3. A présent, A = B, le PGCD des deux entiers donnés vaut 2^k *a.

- Ecrire une fonction PPCM qui retourne le ppcm de deux entiers positifs a et b, tel que :

  PPCM(a, b) * PGCD(a, b) = a * b
- Ecrire une procédure CALCUL permettant de remplir et d'afficher la matrice A en utilisant la fonction PPCM.

Exercice N° 9

Soient deux tableaux de N entiers chacun qui comportent respectivement les numérateurs et les dénominateurs de N fractions.
Etablir un algorithme qui permet de calculer et d'afficher le numérateur et le dénominateur de la somme des N fractions.

Exercice N° 10

Ecrire un algorithme permettant de déterminer et d'afficher tous les facteurs premiers d'un entier positif P donné, qui figurent dans un tableau T de N entiers (N est une constante égale à 10).

Exercice N° 11

Ecrire un algorithme permettant de remplir un tableau T de N éléments entiers (N étant une donnée comprise entre 2 et 50), de déterminer et d'afficher le nombre de diseurs de chaque éléments de T ainsi que la liste de nombres premiers que contient T.

Exercice N° 12

Ecrire un algorithme permettant de déterminer et d'afficher les deux nombres premiers encadrants le PPCM de deux entiers A et B données.

Exercice N° 13

Deux nombres a et b sont jumeaux si :
– a et b sont premiers
– a=b+2
Ecrire un algorithme qui permet de déterminer et afficher les nombres jumeaux compris entre 1 et n (n étant un entier donné).

Exercice N° 14

Deux nombres a et b sont dites étrangers (ou premiers entre eux) si le PGCD (a, b) =1. Ecrire un algorithme permettant de déterminer et d'afficher tous les nombres étrangers compris entre 2 et 100.

Exercice N° 14

Ecrire un algorithme permettant de déterminer et d'afficher le PGCD de des nombres naturels a et b en utilisant les propriétés suivantes:

```
* si a=0:  PGCD(0,b) = b
```

```
  si b=0:  PGCD(a,0) = a
* si a et b sont pairs: PGCD(a,b) = 2 * PGCD(a/2,b/2)
* si a est pair et b impair: PGCD(a,b) = PGCD(a/2,b)
  si a est impair et b pair: PGCD(a,b) = PGCD(a,b/2)
* si a et b sont impairs: PGCD(a,b) = PGCD(a-b,b) si a>b
                          PGCD(a,b) = PGCD(a,b-a) si b>a
```

Exercice N° 15

Ecrire un algorithme permettant de calculer et d'afficher la représentation binaire d'un entier N en octal (base 8).

Exercice N° 16

Ecrire un algorithme permettant de déterminer et d'afficher la représentation binaire d'un caractère C donnée.

Exercice N° 17

Écrire un algorithme qui affiche la suite de tous les nombres parfaits inférieurs ou égaux à un nombre, entier positif donné n. Un nombre est dit parfait s'il est égal à la somme de ses diviseurs stricts.
Exemple : 28 = 1 + 2 + 4 + 7 + 14
Voici la liste des nombres parfaits inférieurs à 10000 : 6, 28, 496, 8128.

Exercice N° 18

Ecrire un algorithme permettant de vérifier et d'afficher si deux entiers donnés A et B sont amis ou non. A et B sont dites amis si la somme des diviseurs de A est égale à B et vice versa.

Exercice N° 19

Ecrire un algorithme permettant de déterminer et d'afficher tous les nombres amis de l'intervalle [1..1000].

Exercice N° 20

Ecrire un algorithme permettant de déterminer et d'afficher le nombre de tours après lesquels deux cyclistes C1 et C2 qui couraient respectivement avec les vitesses V1 et V2 données s'interceptent.

Exercice N° 21

Quelle est la somme des naturels de quatre chiffres qui sont divisibles par 23 et dont la somme des chiffres l'est aussi?

Exercice N° 22

Le crible d'Erathostène, un contemporain d'Archimède, permet de déterminer tous les nombres premiers inférieurs à un nombre entier N.

Rappel: un nombre naturel est dit premier si aucun nombre ne le divise sauf 1 et lui-même. Par convention, 1 n'est pas considéré comme premier.

Voici les différentes étapes de l'algorithme:
Placer d'abord les nombres 1, 2, 3, ..., N dans un tableau.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Commencer avec l'élément en 2ème position et continuer avec les suivants:
si l'élément en position i a été supprimé, passer au suivant sinon, supprimer tous les éléments du tableau (mais pas lui) dont le numéro de position est un multiple de i
Ceci donne: i=2 : l'élément en 2ème position vaut 2. Supprimons donc les éléments en positions 4, 6, ...

1	2	3	x	5	x	7	x	8	x	9	x	11	x	13	x	15	x	19

i=3 : l'élément en 3ème position vaut 3. Supprimons donc les éléments en positions 6, 9, 12, ...
Note: l'élément en position 6 a été supprimé pour la 2ème fois mais ceci ne pose aucun problème.

1	2	3	x	5	x	7	x	9	x	11	x	13	x	x	x	17	x	19

i=4 : l'élément en 4ème position a été supprimé. Ne rien faire.
Comme on peut le Constater, seuls subsistent les nombres premiers.
Ecrire un algorithme qui détermine et affiche les nombres premiers ainsi que les couples jumeaux entre 1 et N (N étant un entier données) en utilisant le principe ci-dessus indiqué.

Exercice N° 23

Ecrire un algorithme qui demande à l'utilisateur un nombre N exprimé dans la base Base et l'exprime dans la base NewBase.

Exercice N° 24

On se propose de crypter un message en utilisant le principe suivant :

1. Chaque caractère de la chaîne sera convertit en sa représentation binaire (8 bits).
2. Appliquer une rotation de pas P donné à la représentation binaire du caractère (P≤8).
3. Remplacer la novelle représentation binaire par le caractère qui lui correspond.

Ecrire un algorithme permettant de saisir une chaîne de caractère CH, de déterminer et d'afficher la chaîne CHC en utilisant le principe de cryptage ci-dessous décrit. Puis de la décrypter (obtenir à nouveau la chaîne initial) et l'afficher.

On souhaite trouver parmi les entiers de l'intervalle [100, 500] les entiers qui admettent le plus grand nombre de diviseurs.
Pour cela écrire un programme permettant de chercher puis d'afficher ces entiers ainsi que le nombre de leurs diviseurs.

Écrire un programme qui permet de lire un entier positif A et de savoir si :

1. A est composé de chiffres formant un palindrome.
2. A est divisible par chacun de ces chiffres.
3. A est divisible par la somme se ses chiffres.

Ecrire un programme permettant d'accomplir les tâches suivantes :
Remplir un tableau T par N entiers positifs (5<N<20).
Pour un élément d'indice p donné, de T, placer dans un autre tableau V :
tous les diviseurs de l'élément d'indice p sauf lui même, s'ils existent dans le tableau T, au début d'un tableau V.
l'élément d'indice p.
tous les multiples de l'élément d'indice p sauf lui même, s'ils existent dans le tableau T, à sa droite.
Remplacer le reste des éléments de V par -1 et l'afficher.
Exemple:
Pour p = 1 et le tableau T suivant :

12	2	0	6	9	48	3	24	10
1	2	3	4	5	6	7	8	9

Le programme affichera le tableau V suivant :

2	6	3	**12**	0	48	24	-1	-1

Écrire un algorithme qui permet de saisir une chaîne de caractères, puis de vérifier et d'afficher s'elle représente un nombre dans une base donnée (entre 2 et 16).

Écrire un algorithme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires.
On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

Dans l'ordre : une chance sur X de gagner

Dans le désordre : une chance sur Y de gagner

X et Y nous sont donnés par la formule suivante, si n est le nombre de chevaux partants et p le nombre de chevaux joués :

$X = n ! / (n - p) !$

$Y = n ! / (p ! * (n – p) !)$