Exercice 1 :

Pour calculer le **sinus** d'un angle **x**, on peut utiliser la formule suivante :

$$\mathbf{sin\ x = 1 - x^3/3! + x^5/5! - x^7/7! + ...}$$

Travail demandé :

Ecrire un algorithme d'une fonction **sinus** qui calcule une valeur approchée de sinus d'un angle x donné à 10^{-6} près. Le calcul s'arrête lorsque la différence entre deux termes consécutifs soit inferieur à 10^{-6}. La valeur approchée de sinus sera alors le dernier terme calculé.

Exercice 2 :

Etant donnés **n, p** deux entiers strictement positifs et $\boldsymbol{C_n^p}$ définie comme suit :

$$\begin{aligned} \mathbf{C_n^0} &= \mathbf{1} \\ \mathbf{C_n^n} &= \mathbf{1} \\ \mathbf{C_n^p} &= \mathbf{C_{n-1}^p + C_{n-1}^{p-1}} \end{aligned}$$

On se propose d'approcher la valeur de S définie par la formule suivante :

$$\mathbf{S = 1 + 2\,\frac{1}{3\,C_2^1} + 2^2\,\frac{1}{5\,C_4^2} + 2^3\,\frac{1}{7\,C_6^3} + \cdots}$$

Travail demandé :

1. En utilisant la définition donnée ci-dessus, écrire un algorithme d'une fonction nommée Combinaison permettant de calculer C_n^p.
2. Utiliser la fonction Combinaison afin d'écrire un algorithme d'un module qui permet de déterminer une valeur approchée de **S** à **epsilon** près.

Exercice 3:

En mathématiques, la constante de **Brun** (B) des nombres premiers jumeaux est la somme de la série des inverses des nombres premiers distants de **2**.

$$B = \left(\frac{1}{3}+\frac{1}{5}\right) + \left(\frac{1}{5}+\frac{1}{7}\right) + \left(\frac{1}{11}+\frac{1}{13}\right) + \left(\frac{1}{17}+\frac{1}{19}\right) + \left(\frac{1}{29}+\frac{1}{31}\right) + \cdots$$

Travail demandé :

Ecrire un algorithme d'une fonction **Brun(epsilon)** permettant de calculer, à epsilon près, une valeur approchée de la constante de Brun définie précédemment (avec epsilon un réel passé en paramètres et dont la valeur est déjà saisie dans le module appelant).

Exercice 4:

Soient **x** un réel positif et ***U*** une suite définie par :

$$\begin{cases} u_0 = \dfrac{(1+x)}{2} \\ u_n = \dfrac{1}{2}\left(u_{n-1} + \dfrac{x}{u_{n-1}}\right) \ pour\ tout\ n > 0 \end{cases}$$

Le terme $\boldsymbol{u_n}$ est une valeur approchée de la racine carrée de x à epsilon prés, si $\left|\frac{u_n - u_{n-1}}{u_{n-1}}\right|$ < **epsilon**

Travail demandé :

1- Quel est l'ordre de récurrence de la suite **U** ? justifiez votre réponse.

2- Ecrire un algorithme d'une fonction **RacineU(x)** qui retourne une valeur approchée de la racine carrée d'un réel positif x à epsilon prés (avec epsilon = 10^{-4}) en utilisant la suite **U** définie précédemment

N.B : Le candidat n'est pas appelé à saisir x

Exercice 5:

Pour évaluer $\mathbf{a^n}$ (a^n = a*a*a*a*…..*a), avec **a** et **n** deux entiers naturels, on a besoin de **n-1** multiplication.
En informatique, l'algorithme d'exponentielle rapide est un algorithme utilisé pour calculer rapidement des grandes puissances entières. Le principe de cet algorithme est basé sur le fait qu'on a :
$\mathbf{a^n}$ =$a^{n/2} * a^{n/2}$ lorsque n est pair et a^n= $a * a^{(n-1)/2} * a^{(n-1)/2}$ lorsque n est impair.
D'où :

$$\mathbf{a^n} = \begin{cases} \mathbf{1} & \textbf{lorsque n=0} \\ \mathbf{a^{n/2} * a^{n/2}} & \textbf{lorsque n est pair} \\ \mathbf{a * a^{(n-1)/2} * a^{(n-1)/2}} & \textbf{lorsque n est impair.} \end{cases}$$

Travail demandé :

1. Ecrire une fonction récursive **Expo_rapide (a,n)** qui permet de calculer $\mathbf{a^n}$ en utilisant le principe décrit précédemment.
2. En faisant appel à la fonction **Expo_rapide (a,n)** de la question **1)**, écrire une fonction **Exponentielle (x)** qui permet de calculer une valeur approchée de $\mathbf{e^x}$ à epsilon prés (avec epsilon = 10^{-5}), sachant que :

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

Avec **n !** représente le factoriel de n

3. Soit la fonction ***f*** définie par :

$$\boldsymbol{f(x)} = \frac{\mathbf{1}}{\boldsymbol{x^n * e^x}}$$

Afin de vérifier que $\lim_{x\to\infty} f(x) = 0$, utiliser les modules définies précédemment pour écrire un algorithme d'une procédure **Verif(Flim, n)** qui permet de stocker dans un fichier texte **Flim**, les valeurs de ***f(x)*** en commençant par **x=1** et en faisant varie x d'un pas égal à **1**. L'écriture dans le fichier **Flim** s'arrête lorsque ***f(x)*** devient inférieure ou égale à $\mathbf{10^{-5}}$

N.B :

- Chaque valeur ***f(x)*** sera stockée dans une ligne du fichier **Flim**
- Le candidat n'est pas appelé à saisir n