|  |  |
| --- | --- |
| *sous-programme* | *[Enseignement secondaire](https://fr.wikipedia.org/wiki/Enseignement_sup%C3%A9rieur)* |

“The computer was born to solve problems that did not exist before.”

*EXERCICE 1:*

Pour **A** et **B** deux entiers strictement positifs, on définit la relation suivante :

Si **(A! \* B! Mod (A + B) = A) OU (A! \* B! Mod (A + B) =B)** Alors **(A+B)** est un nombre premier.

**On dispose d’une matrice de dimension N\*7 pour nous aider à vérifier la relation précédente :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A+B | A! | B! | A! \* B! | A! \* B! Mod (A+B) |
| 2 | 3 | 5 | 2 | 6 | 12 | 12 Mod 5 = 2 |
| 7 | 4 | 11 | 5040 | 24 | 120960 | 120960 Mod 11 = 4 |
| 5 | 2 | 7 | 120 | 2 | 240 | 240 Mod 7 = 2 |
| … | … | … | … | … | … | … |

**Questions :**

1. Ecrire une fonction **somme** qui prend en paramètre deux entiers x et y et retourne x + y
2. Ecrire une fonction **factorielle** qui prend en paramètre un entier n et retourne n!
3. Ecrire une fonction **produit\_f** qui prend en paramètre deux entiers x et y et retourne x! \* y!
4. Ecrire une fonction **pf\_reste** qui prend en paramètre deux entiers x et y et retourne le reste de la division de x! \* y! par x+y
5. Ecrire une fonction **pfr\_verif** qui prend en paramètre deux entiers x et y et qui retourne **vrai** si le reste de la division de x! \* y! par x+y est égale à x ou égale à y, **faux** Sinon.
6. Ecrire une procédure **Remplir\_Relations** qui prend en paramètre un entier **N**, une matrice **M** de dimension **N\*7**, cette procédure nous permet de remplir M avec les différents n couples (A,B) et les résultats de leurs relations comme indiqué dans l'exemple, à l'aide des questions précédentes.
7. Ecrire une procédure **Afficher\_Relations** qui nous permet d’afficher la matrice de relation et de préciser pour chaque ligne si la somme de couple est premier ou pas premier.