CryptoQuest

L'UE de Mathématiques pour la Cryptographie va être découpée en 2 parties :

- une première partie où vous allez voir les bases mathématiques nécessaire avant de passer une partie plus pratique
- une deuxième partie plus pratique, et plus axée sur le domaine de la cryptographie

Prenez connaissance des cours avant d'entamer ce quest !

Lors du quest CryptoQuest, vous allez donc voir quelques notions mathématiques. Ce quest est à faire en mode "piscine" :

- le fichier devra porter le même nom que celui fourni dans le sujet
- l'arborescence demandée sera la même que celle du sujet
- tout non respect de la consigne entraînera un refus de l'étape
- vous savez déjà tout ça !

Le langage à utiliser sera le PHP.
Inutile de vous dire que vous devrez coder à la norme. Un malus sera appliqué sur la note finale en cas d'abus !
Pour chaque étape de ce quest, vous devrez faire une demande de validation afin de pouvoir être corrigé. Attention, nombre limité, utilisez-les avec parcimonie :)

my_modulo_pos

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/my_modulo_pos/

Fichier à rendre : my_modulo_pos.php

Prototype: function my_modulo_pos($int, $n);

Vous devez écrire une fonction prenant en argument deux entiers positifs $int et $n. Vous devrez renvoyer le reste de la division de $int par $n. Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Bien entendu, l'opérateur "modulo" (%) est interdit. Le but est de vous faire comprendre comment marche cet opérateur justement ! Par contre, vous pouvez l'essayer à part pour connaître le résultat attendu.
Petit indice : ce que vous devrez retourner sera forcément compris entre 0 et $n !

my_modulo_neg

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-

MAT2_1_0-1334/CryptoQuest/lagard_v/my_modulo_neg/

Fichier à rendre : my_modulo_neg.php

Prototype: function my_modulo_neg($int, $n);

Vous devez écrire une fonction prenant en argument un entier négatif $int et un entier positif $n. Vous devrez renvoyer le reste de la division de $int par $n. Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant 1).

Bien entendu, l'opérateur "modulo" (%) est interdit. Le but est de vous faire comprendre comment marche cet opérateur justement ! Par contre, vous pouvez l'essayer à part pour connaître le résultat attendu.
Petit indice : ce que vous devrez retourner sera forcément compris entre -$n et 0 !

my_modulo

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/my_modulo/

Fichier à rendre : my_modulo.php

Prototype: function my_modulo($int, $n);

Vous devez écrire une fonction prenant en argument deux entiers (positif ou négatif) $int et $n. Vous devrez renvoyer le reste de la division de $int par $n. Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant 0).

Bien entendu, l'opérateur "modulo" (%) est interdit. Le but est de vous faire comprendre comment marche cet opérateur justement ! Par contre, vous pouvez l'essayer à part pour connaître le résultat attendu.
Petit indice : Juste faire appel à my_modulo_neg et my_modulo_pos ne suffira pas. Pensez à tous les cas possibles !

my_congru

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/my_congru/

Fichier à rendre : my_congru.php

Prototype: function my_congru($a, $b, $n);

Vous devez écrire une fonction prenant en argument trois entiers (positif ou négatif) $a, $b et $n et qui :

- retourne 1 si $a est congru à $b modulo $n
- 0 sinon

Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.

is_classeq

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/is_classeq/

Fichier à rendre : is_classeq.php

Prototype: function is_classeq($tab, $x, $n);

Vous devez écrire une fonction prenant en argument un tableau d'entiers ($tab) et deux entiers $x et $n et qui :

- retourne 1 si tous les éléments de $tab sont de la classe modulo $n de $x
- 0 sinon

Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.

n_classeq

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/n_classeq/

Fichier à rendre : n_classeq.php

Prototype: function n_classeq($tab, $x);

Vous devez écrire une fonction prenant en argument un tableau d'entiers ($tab) et un entier $x et qui :

- retourne le modulo $n de la classe de $x
- 0 sinon

Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli,

tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.

cesar

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/cesar/

Fichier à rendre : cesar.php

Prototype: function cesar($str, $n, $flag);

Vous devez écrire une fonction prenant en argument une chaîne de caractères ($str), deux entiers $n et $flag. Le but de la fonction est la suivante :

- la fonction aura le même comportement que le chiffrement de César
- seuls les caractères alphabétiques seront chiffrés/déchiffrés. Les autres restent tels quels
- un caractère majuscule devient un caractère majuscule
- un caractère minuscule devient un caractère minuscule
- un caractère avec accent sera considéré comme le même caractère sans accent ('é' sera considéré comme 'e' par exemple)
- $n représente la clé de chiffrement/déchiffrement
- $flag représente un flag d'opération : 1 si chiffrement, 0 si déchiffrement

Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.

advanced_cesar

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/advanced_cesar/

Fichier à rendre : advanced_cesar.php

Prototype: function advanced_cesar($str, $n, $alpha, $flag);

Vous devez écrire une fonction prenant en argument une chaîne de

caractères ($str), deux entiers $n et $flag. Le but de la fonction est la suivante :

- la fonction aura le même comportement que le chiffrement de César, mais utilisera un alphabet différent ($alpha)
- seuls les caractères alphabétiques seront chiffrés/déchiffrés. Les autres restent tels quels
- un caractère majuscule devient un caractère majuscule
- un caractère minuscule devient un caractère minuscule
- un caractère avec accent sera considéré comme le même caractère sans accent ('é' sera considéré comme 'e' par exemple)
- $n représente la clé de chiffrement/déchiffrement
- $flag représente un flag d'opération : 1 si chiffrement, 0 si déchiffrement

La différence avec le chiffre de César "classique" est que vous aurez à utiliser un alphabet différent de "abcdefghijklmnopqrstuvwxyz". Vous pourrez spécifier "bzdpxofnhwmglvkjitsdyerqua" par exemple, qui possède les 26 lettres de l'alphabet mais dans le désordre. Le décalage devra alors se faire par rapport à ce nouvel alphabet. Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.

vigenere

Dossier de rendu : https://rendu-svn.etna-alternance.net/v2/2018_Prep'ETNA2_CMG-MAT2_1_0-1334/CryptoQuest/lagard_v/vigenere/

Fichier à rendre : vigenere.php

Prototype: function rev_vigenere($str, $key);

Vous devez écrire une fonction prenant en argument deux chaînes de caractères ($str et $key). Le but de la fonction est la suivante :

- la fonction aura presque le même comportement que le chiffrement de Vigenère. Ici, nous utiliserons la clé de manière différente. En effet, plutôt que de faire boucler la clé de chiffrement à partir du début de la chaîne à chiffrer (comme dans le cours), vous allez le faire à l'envers, à partir de la fin de la chaîne. Exemple en bas du sujet.
- seuls les caractères alphabétiques seront chiffrés/déchiffrés. Les

autres restent tels quels

- un caractère majuscule devient un caractère majuscule
- un caractère minuscule devient un caractère minuscule
- un caractère avec accent sera considéré comme le même caractère sans accent ('é' sera considéré comme 'e' par exemple)
- $key représente la clé de chiffrement.

```
//exemple

//avec vigenere, chaîne "CHIFFRE DE VIGENERE", clé "VIJAI"

CHIFFRE DE VIGENERE
VIJAIVI JA IVIJAIVI

//avec rev_vigenere, chaîne "CHIFFRE DE VIGENERE", clé "VIJAI"
CHIFFRE DE VIGENERE
IVIAJIV IA JIVIAJIV
```

Si jamais il y a un message d'erreur à afficher, vous devrez utiliser la phrase "va t'acheter des doigts !" (avec un saut de ligne pour faire joli, tout en retournant -1).

Toujours et encore, l'opérateur "modulo" (%) est interdit.