

Cryptage et décryptage d'un message

Énoncé

Préliminaire : on se réfère dans ce sujet à un langage de programmation capable de traiter des nombres entiers et des caractères, ce qui est le cas de la plupart des langages y compris ceux que fournissent certaines calculatrices programmables. En informatique, le code ASCII consiste à associer à chaque caractère un code numérique qui est un entier compris entre 0 et 255. Ainsi, le code de @ vaut 64, celui de A est 65, etc.

Questions de syntaxe : dans la plupart des langages de programmation il existe une fonction appelée `chr()` ou `char()` ou `car()` et qui renvoie un caractère à partir de son code ASCII. On entre donc par exemple `chr(65)` pour obtenir la lettre A. La fonction réciproque est souvent nommée `asc()` ou `ord()`, de sorte qu'on tape `ord("A")` ou `asc('A')` (selon le langage) pour obtenir le nombre 65.

Pour simplifier ce qui suit, nous conviendrons de nous limiter à un sous-alphabet formé des lettres majuscules de A à Z et du caractère @ pour marquer les espaces. Dans ces conditions, la formule `ord(c) - 64` renvoie un nombre compris entre 0 et 26 si la variable `c` contient une lettre de notre mini-alphabet.

1. Codage.

- (a) En utilisant le codage décrit ci-dessus, coder le message suivant :

BONJOUR@A@TOUS

On définira un tableau pour ranger les lettres et un autre pour le codage du message.

Appeler l'examineur pour lui montrer l'écran du logiciel après remplissage.

- (b) On va crypter (chiffrer) le message au moyen de la fonction `C` qui, à tout n entier appartenant à $[0;26]$ associe le reste `C(n)` de la division de $13n$ par 27. Adapter la procédure réalisée en 1.(a) pour obtenir les restes `C(n)` correspondant à chaque code n , puis en déduire la lettre correspondante.

Appeler l'examineur pour validation des résultats.

- #### 2. Décodage.
- Notons `D` la fonction qui, à tout entier k appartenant à $[0;27]$, associe le reste de la division de $25k$ par 27. À partir des nombres cryptés trouvés précédemment, retrouver le message originel en utilisant la fonction `D`.

Appeler l'examineur pour vérification du résultat.

- #### 3. Amélioration.
- Le codage proposé ci-dessus est rudimentaire, notamment parce que le caractère d'espacement @ est invariant. On modifie donc la fonction `C` ainsi : `C(n) =` reste de la division de $13n + 8$ par 27. Comment faut-il modifier la fonction `D` ?

Appeler l'examineur pour lui proposer une réponse éventuelle à cette question.

- #### 4. Justification du codage.
- Pour le codage ASCII, deux lettres de l'alphabet sont codées par deux nombres distincts. Il faut donc s'assurer que le cryptage choisi au 1.(b) code deux nombres n et p distincts, compris entre 0 et 26, par deux nombres distincts.

- (a) Montrer que, si $C(n) = C(p)$ alors 27 divise $13(n - p)$.
 (b) En déduire que $n = p$ puis que le codage est valide.

Production demandée

- Écrire le message codé et le message décodé.
- Justifications demandées aux questions 4.(a) et 4.(b).