

Examen blanc du 21/03/2023

Énoncé du problème

Certains codes numériques (les numéros de carte bleue, par exemple) sont organisés de sorte à repérer les erreurs lors de la saisie.

Le principe consiste à produire un nombre "de contrôle" à partir de tous les chiffres du code et à vérifier si ce nombre de contrôle est divisible par 10 : si oui, le code est valide. Sinon, c'est qu'il y a eu une erreur lors de sa saisie.

Pour former ce nombre de contrôle, on multiplie par 2 tous les chiffres du code en **position impaire** (celui le plus à droite étant en position 0, il est en position paire) : si le résultat d'une multiplication donne un chiffre supérieur à 9, on lui soustrait 9. Les chiffres en **position paire** restent comme ils sont.

Par exemple (on utilise ici des codes de 4 chiffres, mais le principe est identique pour des codes de longueur quelconque), le code 1111 devient 2121, tandis que le code 8763 devient 7733 (car $2 \times 6 = 12$, et $12 - 9 = 3$; $2 \times 8 = 16$, et $16 - 9 = 7$).

Puis, on additionne tous les chiffres ainsi obtenus : le code 1111 donne la somme 6 ($2 + 1 + 2 + 1$) et le code 8763 donne la somme 20 ($7 + 7 + 3 + 3$).

Si cette somme est un multiple de 10, le code était correct ; sinon, il était incorrect : le code 1111 était donc incorrect, tandis que le code 8763 était correct.

Travail demandé

1. Écrire l'algorithme par raffinages successifs qui, étant donnée une **chaîne numérique de longueur quelconque** indique si cette suite est correcte. Une chaîne numérique est une chaîne uniquement composée de caractères numériques...
2. Écrire un sous-programme en Go implémentant cet algorithme.
3. Écrire une fonction `main()` demandant de saisir au clavier un chaîne numérique de longueur quelconque, puis affichant si le code représenté par cette chaîne était correct ou non