Devoir d'informatique

D'après Anthony MDF.

Consignes

- Le devoir se fera sur copie double uniquement.
- Le numéro de chaque exercice et de chaque question devra être indiqué sur votre copie.
- Les indentations devront correctement figurer sur votre copie. Vous pourrez par exemple tracer une barre verticale.
- Pour chaque fonction vous donnerez au plus une ligne de commentaire permettant de spécifier votre fonction.

1 Introduction

1.1 Et la marmotte elle met le chocolat dans le papier d'alu

Cette expression provient d'une publicité pour les chocolats Milka datant de la fin des années 1990 où l'on voit des animaux fabriquer du chocolat, dont une marmotte emballer des tablettes de chocolat dans du papier d'aluminium. Le témoin de la scène raconte alors à une femme à la caisse d'un magasin :

— Et alors la marmotte, elle met le chocolat dans le papier d'alu.

Et la femme de répondre, manifestement incrédule :

— Mais bien sûr !

Si on rajoute à cette introduction, une promotion incroyable du type : pour X emballages retournés, on vous redonne une tablette de chocolat, on vient de créer le problème du jour !

1.2 Le problème du jour

Étant donné les trois valeurs suivantes, la tâche consiste à trouver le nombre total de chocolats maximum que vous pouvez manger. Les trois valeurs sont :

- `argent` : argent que vous avez pour acheter des chocolats ;
- `prix` : prix d'un chocolat ;
- `emballage` : nombre d'emballages à retourner pour obtenir un chocolat supplémentaire.

On peut supposer que toutes les valeurs données sont des nombres entiers positifs et supérieurs à 1.

Toutes les questions sont indépendantes. Néanmoins, il est possible de faire appel à des fonctions ou procédures créées dans d'autres questions.

2 Prise en main

2.1 Résolution d'un problème

Pour illustrer le principe, proposons un exemple. Soient les entrées : `argent = 18`, `prix = 2`, `emballage = 2`, elles donnent un nombre de chocolats mangés de 17. En effet, le prix d'un chocolat est de 2, vous pouvez donc acheter 9 chocolats pour une somme de 18. Vous pouvez retourner 8 emballages et en obtenir 4 de plus, il vous reste un emballage. Ensuite, vous avez 5 emballages, vous pouvez retourner de nouveau 4 emballages et en obtenir 2 de plus. Enfin, vous pouvez retourner 2 emballages pour obtenir 1 chocolat supplémentaire. Vous avez alors 2 emballages, le nouveau et celui restant des 9 chocolats, vous pouvez retourner 2 emballages pour obtenir 1 chocolat supplémentaire. Il vous reste un emballage pour une prochaine fois...

Question 1 *Résoudre le problème, pas à pas, comme précédemment avec les entrées* `argent = 15`, `prix = 1`, `emballage = 3`

Question 2 *Justifier que le type* `int` *est un type adapté au traitement de ce problème (par opposition au type* `float`*).*

2.2 Étude d'une suite

Soit la suite $(u_n)_{n\in\mathcal{N},n\geq 0}$ définie par $u_{n+1} = u_n//p + u_n\%p$ avec u_0 et p deux entiers positifs non nuls.

Question 3 *Démontrer que cette suite est une suite d'entiers naturels strictement décroissante pour $p > 1$. Que se passe-t-il dans le cas où $p = 1$?*

3 Résolution du problème

Nous allons résoudre ce problème de différentes manières dans cette section.

3.1 Résolution par fonction récursive

Question 4 *Proposer une fonction récursive* `def chocolatSuppRecursif(nbChocolat:int, emballage:int) -> chocolatSupplementaire:int :` *prenant en argument un nombre de chocolats et le nombre d'emballages nécessaires pour obtenir un chocolat supplémentaire et renvoyant le nombre de chocolats supplémentaires obtenus.*

Question 5 *Proposer une fonction* `def combienTotalChocolat(argent:int, prix:int, emballage:int) -> nbTotalChocolat:int :` *prenant en argument les trois paramètres définis dans la sous-section 1.2 et renvoyant le nombre total de chocolats obtenus.*

Question 6 *Combien d'emballages restera-t-il ? Écrire la (les) ligne(s) de script permettant d'évaluer* `nbEmballage` *pour les trois paramètres définis précédemment.*

Dans le cas où `emballage = 1`, on obtient le message d'erreur suivant `RecursionError: maximum recursion depth exceeded in comparison`.

Question 7 *Pourquoi obtient-on ce message ? Comment évolue le code si on change le* `maximum recursion depth` *?*

3.2 Version itérative

On souhaite maintenant résoudre le problème de manière itérative.

Question 8 *Proposer une fonction itérative* `def combienIteratif(argent:int, prix:int, emballage:int) -> nbTotalChocolat:int :` *prenant en argument les trois paramètres définis dans la sous-section 1.2 et renvoyant le nombre total de chocolats obtenus.*

4 Représentation des nombres

Soit un sytème d'exploitation ou les nombres sont codés sur 8 bits.

Question 9 *Combien d'entiers positifs sont codables ? Quel est le plus petit entier codable ? Quel est le plus grand ?*

Question 10 *On souhaite coder des entier positifs et négatifs. Quel est le plus petit entier codable ? Quel est le plus grand ?*

Question 11 *Coder le nombre 23 en binaire. Coder le nombre −23 en binaire.*

Question 12 *Coder 101100 en décimal.*

Question 13 *Coder 101100 en hexadécimal.*

Question 14 *Coder A9 en décimal.*