

| | | |
|--|---|---|
|  Région académique | « Défi algorithmique et programmation » <i>de l'IREM des Antilles</i> - Collège et Lycée - décembre 2019: <i>fiche élève</i> |  |
|--|---|---|

« Le défi algorithmique et programmation » de l'IREM, vous est proposé dans le cadre du partenariat entre l'IREM (Institut de Recherche sur l'Enseignement des Mathématiques) de l'Université des Antilles et le Rectorat de l'académie de Guadeloupe.

| | | | | | | | |
|-------------|---------|---|-----------|---|-------------|---------------|---|
| Catégorie : | Logique | X | Numérique | X | Géométrique | Algorithmique | X |
|-------------|---------|---|-----------|---|-------------|---------------|---|

ENONCÉ

« Défi algorithmique et programmation », décembre 2019

« L'automate »



Vous observerez que de plus en plus de commerces sont équipés d'automates qui encaissent et rendent la monnaie. Ces automates présentent l'avantage d'être hygiéniques et de décourager les voleurs, car l'argent inséré dans la caisse se déverse dans un coffre-fort inviolable. Ces appareils rendent la monnaie de façon optimale, c'est-à-dire avec le nombre minimum de billets et de pièces.

Par exemple, pour 76 €, l'automate pourrait vous remettre :

1 billet de 50 € , 2 billets de 10€, et 6 pièces de 1€ soit total 9 (billets et pièces).

ou

Afin de remettre un nombre minimal de pièces et de billets, l'automate vous rendra :

1 billet de 50 € , 1 billet de 20€, 1 billet de 5€ et 1 pièce de 1€ soit un total 4 (billets et pièces).

Le défi de ce mois est de réaliser un programme en Scratch (en collège) ou en Python (en lycée) qui affiche le nombre optimal (minimal) de billet(s) et de pièce(s) pour atteindre une somme à remettre. Pour ce défi, la somme à remettre sera une valeur entière.





Région académique

« Défi algorithmique et programmation »
de l'IREM des Antilles
- Lycée -

Novembre 2019



« Le défi algorithmique et programmation » de l'IREM, vous est proposé dans le cadre du partenariat entre l'IREM (Institut de Recherche sur l'Enseignement des Mathématiques) de l'Université des Antilles et le Rectorat de l'académie de Guadeloupe.

| | | | | | | | | |
|-------------|---------|---|-----------|---|-------------|--|---------------|---|
| Catégorie : | Logique | X | Numérique | X | Géométrique | | Algorithmique | X |
|-------------|---------|---|-----------|---|-------------|--|---------------|---|

ENONCÉ

« Défi algorithmique et programmation », décembre 2019

« L'automate »



Vous observerez que de plus en plus de commerces sont équipés d'automates qui encaissent et rendent la monnaie. Ces automates présentent l'avantage d'être hygiéniques et de décourager les voleurs, car l'argent inséré dans la caisse se déverse dans un coffre-fort inviolable. Ces appareils rendent la monnaie de façon optimale, c'est-à-dire avec le nombre minimum de billets et de pièces.

Par exemple, pour 76 €, l'automate pourrait vous remettre :

1 billet de 50 € , 2 billets de 10€, et 6 pièces de 1€ soit total 9 (billets et pièces).

ou

Afin de remettre un nombre minimal de pièces et de billets, l'automate vous rendra :

1 billet de 50 € , 1 billet de 20€, 1 billet de 5€ et 1 pièce de 1€ soit un total 4 (billets et pièces).

Le défi de ce mois est de réaliser un programme en Scratch (en collège) ou en Python (en lycée) qui affiche le nombre optimal (minimal) de billet(s) et de pièce(s) pour atteindre une somme à remettre. **Pour ce défi, la somme à remettre sera une valeur entière.**

Pour aller plus loin : [Problème du rendu de monnaie](#) (wikipédia)



| Niveau | Cycle 4 à Lycée | | | | | |
|--|---|---|--------------------------------|---|-------------|---|
| Champ disciplinaire | Nombres et calcul | X | Géométrie | | Fonctions | |
| | Statistiques et probabilités | | Algorithmique et programmation | X | | |
| Compétences mathématiques travaillées | Chercher | X | Modéliser | X | Représenter | X |
| | Raisonner | X | Calculer | X | Communiquer | |
| Connaissances et compétences associées | Résoudre des problèmes motivants et substantiels, afin de stabiliser connaissances, méthodes et stratégies. | | | | | |
| Compétences visées pour la mise en œuvre: | <p>CHERCHER:</p> <ul style="list-style-type: none"> • prélever et organiser les informations nécessaires à la résolution de problèmes à partir de supports variés : textes; • s'engager dans une démarche, observer, questionner, manipuler, expérimenter, émettre des hypothèses, en mobilisant des outils ou des procédures mathématiques déjà rencontrées, en élaborant un raisonnement adapté à une situation nouvelle; • tester, essayer plusieurs pistes de résolution; • expérimenter – en particulier à l'aide d'outils logiciels; • décomposer un problème en sous-problèmes. <p>MODELISER:</p> <ul style="list-style-type: none"> • traduire en langage mathématique une situation réelle (par exemple à l'aide d'équations, de fonctions, de configurations géométriques, d'outils statistiques); • faire une simulation, valider ou invalider un modèle; <p>REPRESENTER:</p> <ul style="list-style-type: none"> • adapter pour traiter un problème ou pour étudier un objet mathématique; • produire et utiliser plusieurs représentations des nombres; • choisir un cadre (numérique, algébrique, géométrique...), changer de registre. <p>RAISONNER:</p> <ul style="list-style-type: none"> • démontrer, trouver des résultats partiels et les mettre en perspective. <p>CALCULER:</p> <ul style="list-style-type: none"> • appliquer des techniques et mettre en œuvre des algorithmes. | | | | | |

| | Analyse de la situation | |
|--|--|--|
| | <p>Ce problème est similaire à celui consistant à atteindre un poids donné en utilisant des boîtes à poids (Photo ci-contre). Pour cela le commerçant commence par le poids le plus gros ne dépassant pas le poids de ce qu'il a à peser puis poursuit décroissante.</p> <p>Ainsi pour rendre la monnaie avec un nombre minimal de pièces et de billets, il faut faire en sorte que la somme à atteindre soit « le plus porté » par les sommes plus élevées.</p> |  |
| | Principe de l'algorithme | |
| | <p>L'algorithme consistera dans un premier temps à ranger les valeurs à utiliser dans une structure de donnée appelée liste et dans un second temps à positionner dans une deuxième liste les effectifs associés aux valeurs utilisées.</p> <p>Ainsi nous allons préparer les deux listes suivantes :</p> <ol style="list-style-type: none"> 1. la liste des valeurs 50€, 20€, 10€, 5€, 2€, 1€, rangées dans l'ordre décroissant. 2. la liste des effectifs associés aux différentes valeurs utilisées. | |
| | Notion de liste dans les documents d'accompagnement | |
| <p>De l'analyse de la situation, à la résolution algorithmique du problème.</p> | <p>Au collège : Ce problème peut se résoudre sans avoir à utiliser des listes mais il sera alors beaucoup plus long car il nécessitera d'effectuer un test pour chacune des valeurs (50€, 20€, 10€, 5€, 2€, 1€) . Le programme de collège ne fait pas référence au liste toutefois le document d'accompagnement: Algorithmique et programmation en cycle 4, fait référence à l'utilisation de liste et stipule que l'utilisation de cette structure de donnée, paraît indispensable dans certaines d'activités ou de projets. Il a été mentionné également que certaines activités ne demandent qu'une utilisation très élémentaire d'une liste, et peuvent donc servir d'introduction à cette notion. Ce défi peut y contribuer.</p> <p>Au lycée : La notion de liste est explicitement inscrite dans le programme de 1ère toutefois le document d'accompagnement : Ressources pour le lycée mathématiques (série générale) : « Algorithmique et programmation » mentionne que les listes ne font pas l'objet d'un enseignement spécifique en seconde, mais peuvent être utilisées comme objets simples et pratiques.</p> | |

Structure de l'algorithme

On demande à l'utilisateur de saisir, la valeur « s » correspondante à la monnaie qui va être remise au client.

On parcourt la liste des valeurs (*de la plus grande valeur à la plus petite valeur*) :

Tant que $s \neq 0$ faire

Si $s \geq$ la valeur concernée alors

faire :

Incrémenter l'effectif associé à la valeur concernée.

$s \leftarrow s -$ valeur concernée (*la nouvelle somme à remettre devient : la somme initiale moins la valeur du billet ou de la pièce*)

sinon

Passer la valeur suivante (*valeur de montant inférieur*)

Affichage du résultatat :

Afficher chacune des valeurs et l'effectif associé.

« LYCEE »

Proposition de solution du défi avec Python

| Version texte (à copier - coller) | Description |
|---|--|
| <pre>s=int(input("Somme à remettre :")) print(s) l=[50,20,10,5,2,1] r=[0]*6 print(l) print(r) i=0 while s!=0: if s >= l[i]: r[i]=r[i]+1 s=s-l[i] else: i=i+1 print("La machine automatique rendra :") for i in range(6): if r[i]!=0: if l[i]>=5: print(r[i]," billet(s) de",l[i]," euros") else: print(r[i]," pièce(s) de",l[i]," euros")</pre> | <p><i>saisie de la somme entière</i> <i>affichage la somme à remettre s</i> <i>liste des billets et pièces utilisés</i> <i>liste des effectifs associés: $r=[0]^*6=r=[0,0,0,0,0,0]$</i> <i>affichage la liste l</i> <i>affichage de la liste r</i> <i>initialisation de l'indice des valeurs à 0</i> <i>Tant que $s \neq 0$ faire :</i> <i> Si la somme s est supérieure ou égale à valeur associée à l'indice i alors faire :</i> <i> *Ajouter 1 à l'effectif correspondant à la valeur d'indice i.</i> <i> *s prend la valeur s moins la valeur d'indice i .</i> <i> Sinon passer à la valeur suivant</i> <i>Afficher ("La machine automatique rendra :")</i> <i>Pour i allant de 0 à 5 faire :</i> <i> Si l'effectif associé à la valeur 0 :</i> <i> *si la valeur concernée est supérieure ou égale à 5 alors</i> <i> afficher l'effectif associé à la valeur, suivi de « billet(s) »</i> <i> sinon afficher l'effectif associé à la valeur suivi de « pièce(s) »</i></p> |

Copie d'écran

```
main.py  ⏺ saved

1  s=int(input("Somme à remettre :"))
2  print(s)
3  l=[50,20,10,5,2,1]
4  r=[0]*6
5  print(l)
6  print(r)
7  i=0
8  while s!=0:
9      if s >= l[i]:
10         r[i]=r[i]+1
11         s=s-l[i]
12     else:
13         i=i+1
14 print("La machine automatique rendra :")
15 for i in range(6):
16     if r[i]!=0:
17         if l[i]>=5:
18             print(r[i]," billet(s) de",l[i]," euros")
19         else:
20             print(r[i]," pièce(s) de",l[i]," euros")
21
```

exemple

Vous pouvez tester le script ci-dessus sur [Python 3 - repl.it](#) (site permettant de faire du python 3 en ligne sans installation) en faisant un Copier - Coller dans la fenêtre centrale puis lancer le programme en cliquant en haut sur RUN.

Il faut vérifier après le collage la mise en forme (l'indentation, les guillemets qui souvent sont à retaper, ...) car sinon vous aurez des messages d'erreurs.

Prenons l'exemple d'une somme à remettre de : 69 €

Le résultat obtenu par le programme est : $1 \times 50 + 1 \times 10 + 1 \times 5 + 2 \times 2 = 69$ €

affichage obtenu

```
Somme à remettre :69
69
[50, 20, 10, 5, 2, 1]
[0, 0, 0, 0, 0, 0]
La machine automatique rendra :
1 billet(s) de 50 euros
1 billet(s) de 10 euros
1 billet(s) de 5 euros
2 pièce(s) de 2 euros
> |
```

« COLLEGE »

Proposition de solution du défi avec SCRATCH 3

Dans un premier temps, nous allons créer deux listes et deux variables :

Les listes

- monnaie : qui représente les sommes qui peuvent, être remises
- effectif monnaie : les effectifs des sommes remises

| monnaie | effectif monnaie |
|----------------|------------------|
| 1 50 | 1 1 |
| 2 20 | 2 0 |
| 3 10 | 3 1 |
| 4 5 | 4 1 |
| 5 2 | 5 0 |
| 6 1 | 6 0 |
| + longueur 6 = | + longueur 6 = |

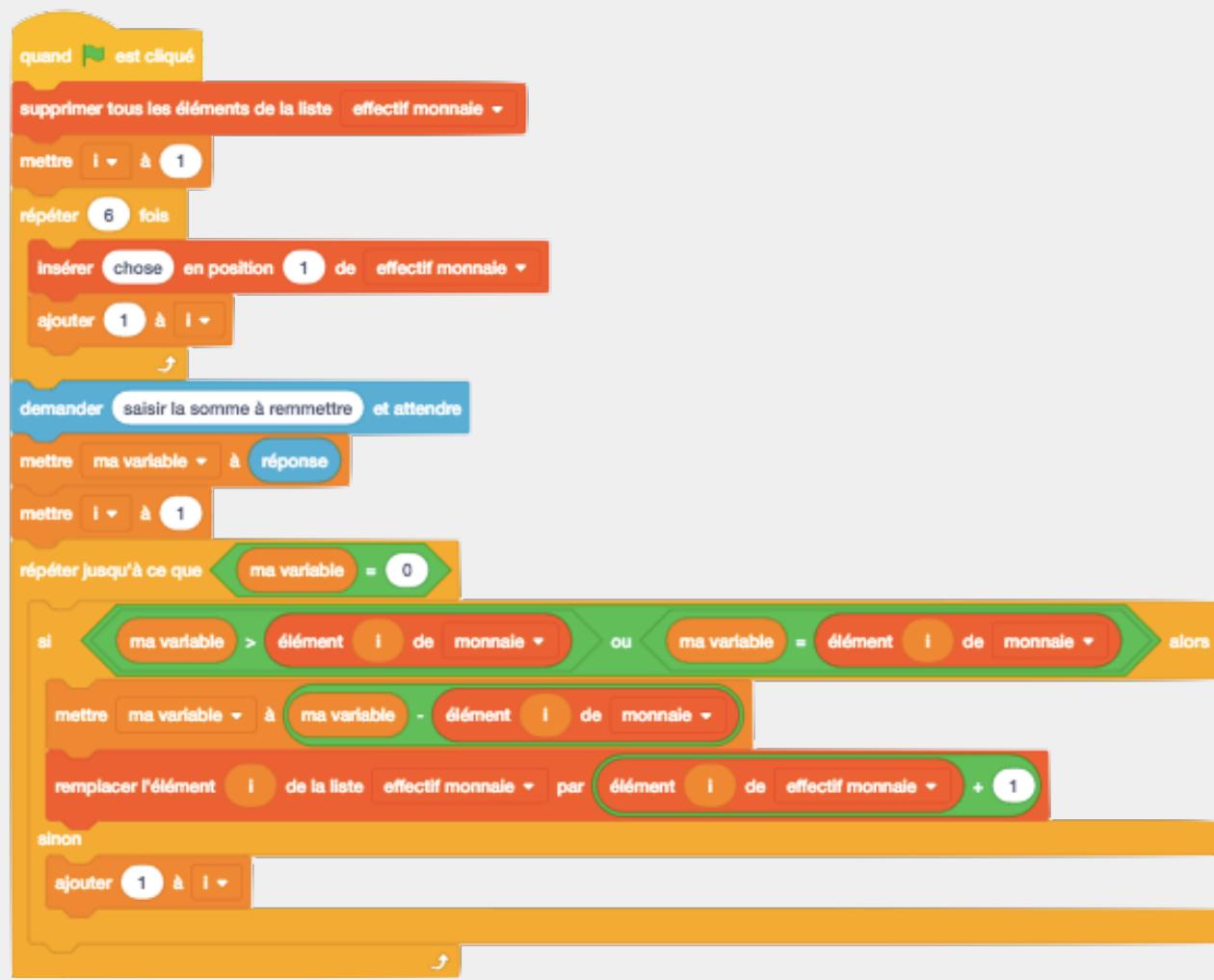
Les variables:

- i : correspond à la ligne dans les listes
- ma variable: est la somme restant à rendre

Créer une variable



le programme



exemple

Affichage obtenu

| monnale | effectif monnale |
|----------------|------------------|
| 1 50 | 1 1 |
| 2 20 | 2 chose |
| 3 10 | 3 1 |
| 4 5 | 4 1 |
| 5 2 | 5 2 |
| 6 1 | 6 chose |
| + longueur 6 = | + longueur 6 = |

Prenons l'exemple d'une somme à remettre de :
69 €

Le résultat obtenu par le programme est :
 $1 \times 50 + 1 \times 10 + 1 \times 5 + 2 \times 2 = 69$ €

| Eléments de différenciation et exploitations possibles | |
|---|--|
| Rappels des contenus des programmes sur les attendus – Cycle 4 | Attendus de fin de cycle - écrire, mettre au point et exécuter un programme simple. Ecrire, mettre au point, exécuter un programme. Connaissances : notions d'algorithme et de programme ; notion de variable informatique ; déclenchement d'une action par un événement ; séquences d'instructions, boucles, instructions conditionnelles. Compétences associées : écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. |
| Passage de scratch à Python en lycée | La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Il s'agira d'aider les élèves à réussir le passage de Scratch à Python . |
| Piste de prolongement | Modifier le programme pour travailler sur la notion de base. Voir comment on peut utiliser les principes de l'algorithmique créé pour écrire un nombre entier en binaire ou dans une base quelconque. |
| Piste de différenciation | Inscrire les apprentissages de l'algorithmique dans la progressivité du cycle 3 au cycle 4 puis du cycle 4 au lycée. Proposer différents niveaux pour ce défi. On pourra commencer avec un nombre limité de billets et de pièces puis traiter tous les cas de figures avec notamment des sommes avec des valeurs décimales. Ce défi peut se transformer en jeu consistant à trouver une somme donnée (avec le minimum de billets et pièces) à partir de billets et de pièces disponibles. |

Situation 1: « la borne d'information automatique »



Autres pistes

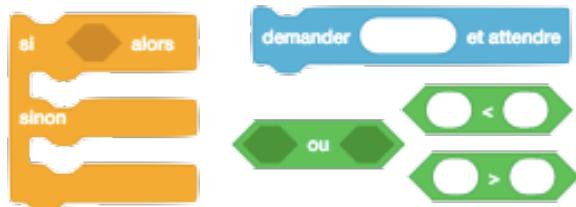
A l'entrée du cinéma se trouve une borne qui vous donne le montant à payer en fonction de votre âge.

Si vous avez moins de 14 ans ou plus de 70 ans vous payerez 4,50€.

Pour les autres le tarif est de 8€.

Proposer un programme qui à partir de l'âge du client lui annonce le prix qu'il va payer.

Différenciation possible en collège:
proposer certains blocs qui seront utiles.



- Lycée - Python -

Vous pouvez tester le script ci-dessus sur [Python 3 - repl.it](#) en faisant un Copier/Coller Il faut vérifier après le collage la mise en forme (l'indentation, les guillemets qui souvent sont à retaper, ...) car sinon vous aurez des messages d'erreurs.

Script

```
reponse=int(input("Quel est votre âge ? "))
if (reponse<14 or reponse>70):
    print("Le prix à payer est de 4.50 €")
else:
    print("Le prix à payer est de 8 €")
```

Aperçu sur [Python 3 - repl.it](#)

```
main.py      saved
1  reponse=int(input("Quel est votre âge ? "))
2  if (reponse<14 or reponse>70):
3      | | | | print("Le prix à payer est de 4.50 €")
4  else:
5      | | | | print("Le prix à payer est de 8 €")
```

Proposition de solution

situation 1

Affichage obtenu

```
Quel est votre âge ? 24
Le prix à payer est de 8 €
> █
```

- Collège - SCRATCH -



la réponse à ce bloc est stockée par défaut dans le bloc « réponse » le bloc « ou », permet de faire deux tests

Situation 2: « la balance automatique »



Autres pistes

Dans les rayons fruits et légumes des supermarchés on trouve des balances automatiques qui éditent les étiquettes pour chaque denrées achetées et pesées par l'appareil.

Ici, on vous demande d'écrire un programme qui dans un premier temps propose une masse en kilogrammes de « tomate cerise » comprise entre 100 grammes et 5 kilogrammes et l'afficher.

Puis dans un second temps le programme doit annoncer le prix à payer au centime près, sachant que le prix des tomates cerises est de 4,14€ pour un kilogramme.

Différenciation possible en collège:
proposer certains blocs qui seront utiles.

nombre aléatoire entre et
arrondi de

- Lycée - Python -

Vous pouvez tester le script ci-dessous sur [Python 3 - repl.it](#) en faisant un Copier/Coller. Il faut vérifier après le collage la mise en forme (l'indentation, les guillemets qui souvent sont à retaper, ...) car sinon vous aurez des messages d'erreurs.

Script

```
from random import randint
poids=randint(100,5000) # poids aléatoire en grammes
poids=poids/1000 # conversion du poids en kg
print(" vous achetez ", poids , "kg de tomates cerises. ")
prix=poids*4.14 #calcul du prix
print(" vous devez payer ", round(prix,2), " €" ) # affichage du prix au centime près
```

Aperçu sur [Python 3 - repl.it](#)

Proposition de solution

situation 2

The screenshot shows a code editor window for Python 3. The file is named 'main.py'. The code is identical to the one above, with syntax highlighting for keywords like 'from', 'import', and 'print'. The code is saved. Below the code, the output of the script is shown in a terminal-like interface. The output text is:
Vous achetez 0.371 kg de tomates cerises.
vous devez payer 1.54 €
▶ []

Affichage obtenu

```
Vous achetez 0.371 kg de tomates cerise
s.
vous devez payer 1.54 €
▶ [ ]
```

- Collège - SCRATCH -

Remarque: nous choisissons d'utiliser deux variables, ce qui n'est pas forcément nécessaire pour résoudre ce problème. Mais il est important que les élèves s'habituent à utiliser pour être prêt pour le lycée.

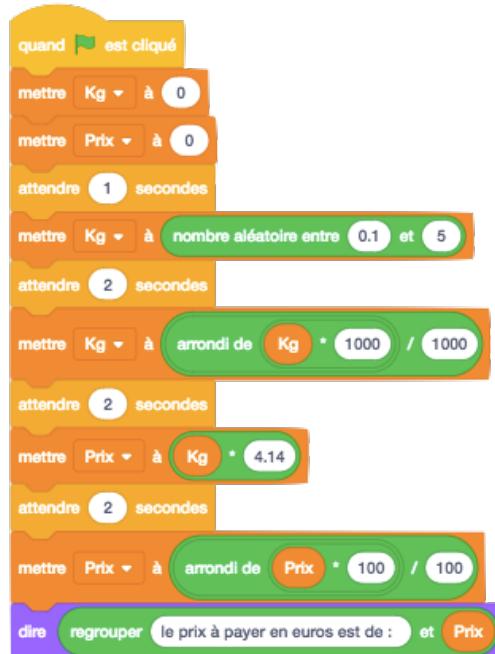
Les variables :

Proposition de solution

situation 2



remarque : nous faisons le choix d'un programme « pas à pas » avec temps d'attente entre chaque étape pour les collégiens afin que tout soit identifiable et puisse être expliquée



initialisation des deux variables « Kg » et « Prix »

*détermine le poids de façon aléatoire
permet d'arrondir le poids au gramme près*

calcul du prix à payer, ce prix n'est pas en centime

*ici on arrondi le « Prix » au centime près.
permet d'afficher deux objets. Ici on affiche un texte et une valeur*