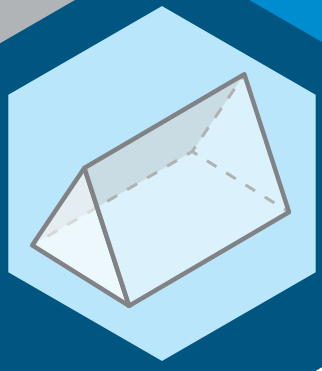
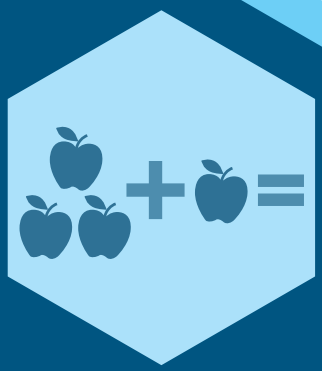
7e année

En avant, les maths!

Une approche renouvelée pour l'enseignement et l'apprentissage des mathématiques

MINILEÇON





ALGÈBRE

Créer et modifier des codes comprenant des événements influencés par un dénombrement prédéfini et/ou un sous-programme et d'autres structures de contrôle

RÉSUMÉ

Dans cette minileçon, l'élève crée et modifie des codes comprenant des événements influencés par un dénombrement prédéfini, un sous-programme et d'autres structures de contrôle.

PISTES D'OBSERVATION

L'élève :

- crée des représentations de situations mathématiques de façon computationnelle en écrivant et exécutant des codes efficaces;
- écrit et exécute des codes comprenant des événements influencés par un dénombrement prédéfini;
- écrit et exécute des codes comprenant des sous-programmes ou autres structures de contrôle;
- lit et modifie des codes donnés et décrit l'incidence de ces changements sur les résultats et l'efficacité.

MATÉRIEL

- un logiciel de programmation par blocs gratuit.

CONCEPTS MATHÉMATIQUES

Le concept mathématique nommé ci-dessous sera abordé dans cette minileçon. Une explication de celui-ci se trouve dans la section **Concepts mathématiques**.

Domaine d'étude	Concept mathématique
Algèbre	Création et modification de codes en situation de résolution de problèmes

PARTIE 1 – EXPLORATION GUIDÉE

Déroulement

- Consulter, au besoin, la fiche **Création et modification de codes en situation de résolution de problèmes** de la section **Concepts mathématiques** afin de revoir avec les élèves comment créer et modifier des codes comprenant des sous-programmes et des dénombrements prédéfinis ainsi que la terminologie liée à ces concepts en vue de les aider à réaliser l'activité. Il importe de ne pas présenter le code complété et fonctionnel. Les élèves doivent découvrir comment le faire et le modifier dans cette minileçon.

- Présenter aux élèves l'**Exemple 1**, soit créer un code contenant un dénombrement prédéfini et un sous-programme permettant de faire tourner un dé un certain nombre de fois et de lire et modifier un code pour faire tourner un dé à 20 faces.

- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour écrire et exécuter des codes efficaces et pour modifier un code donné.

- Demander à quelques élèves de faire part au groupe-classe de leur solution et d'expliquer les stratégies utilisées pour créer un code et modifier un code donné. Inviter les autres élèves à poser des questions afin de vérifier leur compréhension.

- À la suite des discussions, s'assurer que les élèves établissent des liens entre la tâche de création et la modification de codes.

  Note : Au besoin, consulter le corrigé de la partie 1 pour obtenir des exemples de stratégies.

- Encourager les élèves à améliorer leur travail en y ajoutant les éléments manquants.

- Au besoin, présenter aux élèves l'**Exemple 2**, soit créer un code contenant un dénombrement prédéfini et un sous-programme permettant de faire tourner une pièce de monnaie 10 fois et modifier un code donné de façon à ce que l'utilisatrice ou l'utilisateur puisse choisir le nombre de fois qu'elle ou qu'il désire faire tourner une pièce de monnaie.

CORRIGÉ

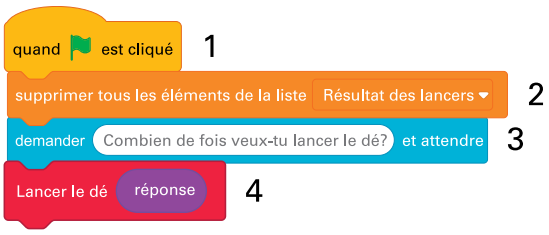
EXEMPLE 1

a) À l'aide d'un logiciel de programmation de ton choix, crée un code contenant un dénombrement prédéfini et un sous-programme permettant de faire rouler un dé un certain nombre de fois.

STRATÉGIE

Écrire et exécuter des codes efficaces comprenant un dénombrement prédéfini et un sous-programme

À l'aide d'un logiciel de programmation par blocs tel que Scratch, je crée un code pour faire rouler un dé un certain nombre de fois. Afin de permettre à l'utilisatrice ou à l'utilisateur de choisir le nombre de fois qu'elle ou qu'il désire faire rouler le dé, je crée un petit ensemble d'instructions, soit un sous-programme. De plus, pour m'assurer que l'utilisatrice ou l'utilisateur peut faire rouler le dé plus d'une fois, j'ajoute un dénombrement prédéfini de façon à ce que les instructions soient répétées.



1. Je débute ma séquence en intégrant un bloc d'événement « Quand le drapeau vert est cliqué ».

2. Sous la catégorie variable, je crée une liste que je nomme « Résultat des lancers ». J'ajoute un bloc « Supprimer tous les éléments de la liste Résultat des lancers » de façon à réinitialiser les données chaque fois que l'utilisatrice ou l'utilisateur souhaite utiliser le programme.

3. J'utilise le bloc capteur « Demander… et attendre » afin de poser la question suivante à l'utilisatrice ou l'utilisateur : Combien de fois souhaites-tu lancer le dé?

4. Je crée un bloc « Lancer le dé ».

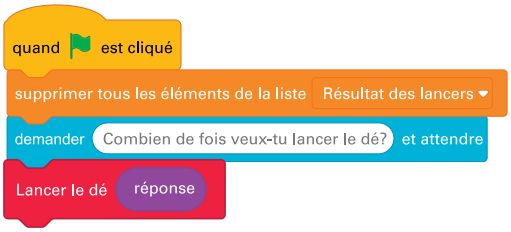
5. Sous la catégorie « Mes blocs », je crée un bloc que je nomme « Lancer le dé » avec une entrée que je nomme « Nombre de lancers ». Ceci est le début de mon sous-programme. Le sous-programme permettra à l'utilisatrice ou l'utilisateur d'entrer le nombre de lancers souhaités.

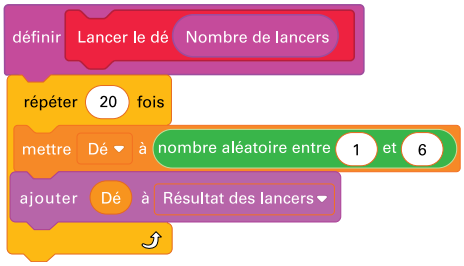
6-7-8.

Je crée une variable que je nomme « Dé » ainsi qu'une liste que je nomme « Résultats des lancers ».

J'ajoute un dénombrement prédéfini de façon à ce que le dé soit lancé plus d'une fois. Je m'assure ainsi que la boucle est répétée selon le nombre de lancers choisi par l'utilisatrice ou l'utilisateur. Dans mon bloc de contrôle « Répéter », j'insère une variable « Mettre Dé à nombre aléatoire entre 1 à 6 ». Ce nombre aléatoire est ensuite enregistré dans la variable « Dé ». À chaque itération de la boucle, le résultat du dé est inscrit dans la liste.

b) Voici un code qui offre la possibilité à l'utilisatrice ou à l'utilisateur de déterminer le nombre de fois qu'elle ou qu'il souhaite lancer un dé à 20 faces. Cependant, le code comporte 2 erreurs. Peux-tu les identifier? Modifie le code et décris l'incidence de ces changements sur les résultats et l'efficacité.

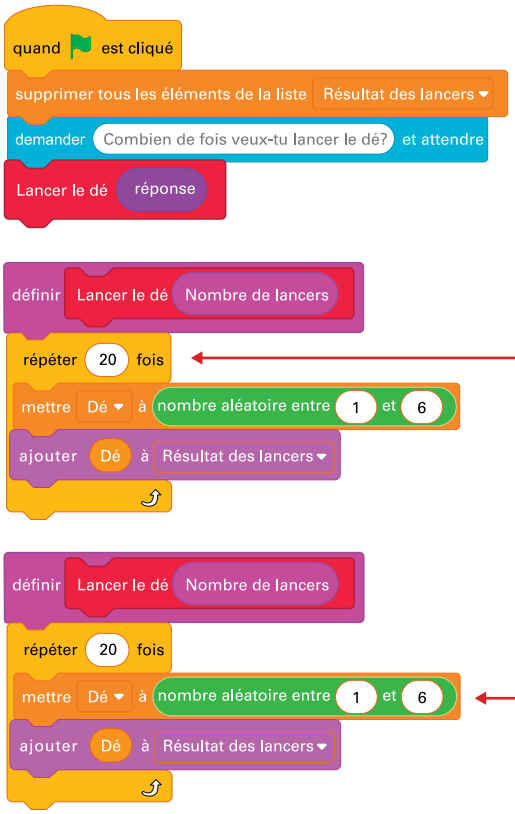




STRATÉGIE

Faire des prédictions en lisant un code et modifier un code donné

Je fais des prédictions pour déterminer les 2 erreurs que comporte le code donné.



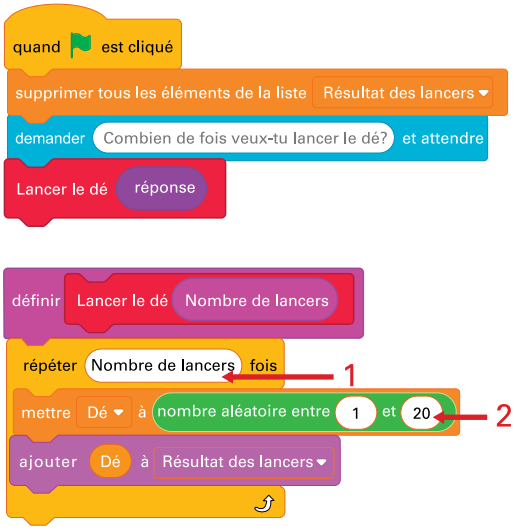
Prédiction 1

En observant le code donné, je remarque que le dé est toujours lancé 20 fois. Même si l'utilisatrice ou l'utilisateur entre un nombre de lancers, le dé sera toujours lancé 20 fois. Je dois donc modifier le code de manière à ce que l'utilisatrice ou l'utilisateur puisse choisir le nombre de lancers qu'elle ou qu'il désire exécuter.

Prédiction 2

Je remarque que le nombre aléatoire du dé se situe entre 1 et 6. Je dois modifier le code pour obtenir un nombre aléatoire entre 1 et 20 puisqu'il s'agit d'un dé à 20 faces.

Je modifie ensuite le code pour déterminer si mes prédictions sont justes et si je dois effectuer d'autres changements :



En modifiant le code, j'observe que le résultat attendu est atteint. L'utilisatrice ou l'utilisateur peut maintenant déterminer le nombre de fois qu'elle ou qu'il désire lancer le dé à 20 faces.

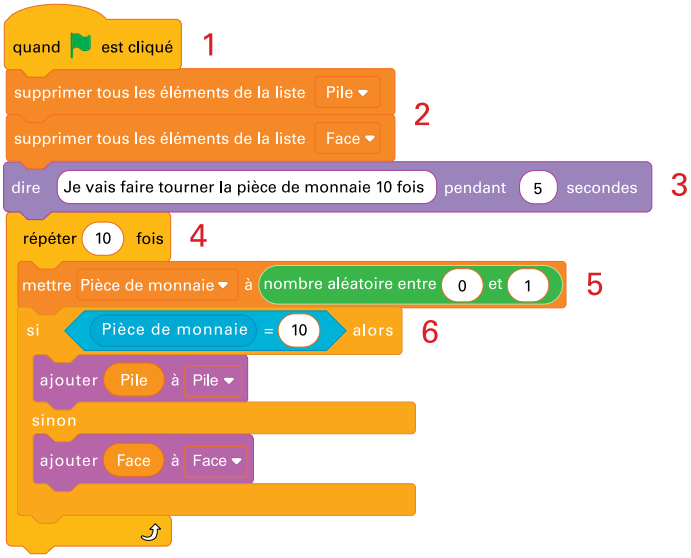
EXEMPLE 2

a) À l'aide d'un logiciel de programmation de ton choix, crée un code qui fait tourner une pièce de monnaie 10 fois à l'aide d'un dénombrement prédéfini.

STRATÉGIE

Écrire et exécuter un code efficace à l'aide d'un dénombrement prédéfini et d'un sous-programme

À l'aide d'un logiciel de programmation par blocs tels que Scratch, je crée un code de façon à faire tourner une pièce de monnaie 10 fois. J'intègre un dénombrement prédéfini de façon à ce que les instructions soient répétées 10 fois.

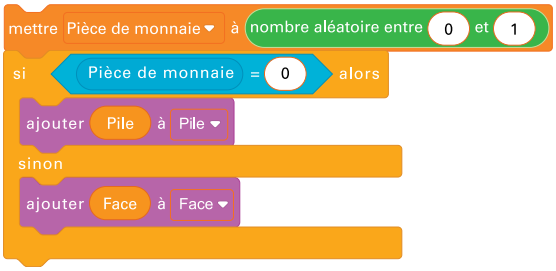


1. Je débute ma séquence en intégrant un bloc d'événement « Quand le drapeau vert est cliqué ».

2. Sous la catégorie « Variable », je crée deux listes que je nomme « Pile » et « Face ». J'ajoute les blocs « Supprimer tous les éléments de la liste Pile » et « Supprimer tous les éléments de la liste Face » de façon à réinitialiser les données chaque fois que l'utilisatrice ou l'utilisateur souhaite utiliser le programme.

3. J'utilise le bloc d'apparence « Dire… pendant cinq secondes » afin de spécifier que la pièce sera lancée 10 fois.

4-5-6.
J'ajoute un bloc de contrôle « Répéter » afin que la pièce de monnaie puisse tourner 10 fois et pour obtenir un résultat aléatoire entre pile ou face.



Dans mon bloc de contrôle « Répéter », j'ajoute la variable « Mettre pièce de monnaie à nombre aléatoire entre 0 et 1 ». J'utilise le bloc d'opérateur « Nombre aléatoire entre 0 et 1 » (0 représente pile et 1 représente face). Ces nombres aléatoires sont ensuite enregistrés dans la variable « Pièce de Monnaie ». J'utilise ensuite le bloc de contrôle « si <> alors, sinon ». Si le nombre choisi est 0, ce sera pile, ALORS il s'ajoute à la liste « Pile ». SINON, ce sera forcément face puisqu'il s'agit de la seule autre possibilité, donc il s'ajoute à la liste « Face ».

b) Modifie ton code de façon à ce que l'utilisatrice ou l'utilisateur puisse choisir le nombre de fois qu'elle ou il désire faire tourner la pièce de monnaie en y insérant un sous-programme.

STRATÉGIE

Modifier un code en y insérant un sous-programme.

Afin que l'utilisatrice ou l'utilisateur puisse choisir le nombre de fois qu'elle ou qu'il désire faire tourner la pièce de monnaie, je dois intégrer un sous-programme.

1. Je débute ma séquence en intégrant un bloc d'événement « Quand le drapeau vert est cliqué ».

2. Sous la catégorie « Variables », je crée deux listes que je nomme « Pile » et « Face » et une variable qui se nomme « Pièce de monnaie ». Je réinitialise toutes les informations qui sont présentes dans les listes en choisissant le bloc « supprimer tous les éléments de la liste « Pile » » et le bloc « supprimer tous les éléments de la liste « Face ».

3. J'utilise le bloc de capteur « Demander » afin de savoir combien de fois la pièce devra être lancée.

4. J'ajoute le capteur « réponse » afin que la pièce soit lancée selon le nombre indiqué par l'utilisatrice ou l'utilisateur.

5. Sous la catégorie « Mes blocs », je crée un bloc que je nomme « Lancer pièce » avec une entrée que je nomme « Nombre de lancers de pièce ». Ceci est le début de mon sous-programme. Le sous-programme permettra à l'utilisatrice ou l'utilisateur d'entrer le nombre de lancers souhaités.

6. Je glisse l'entrée « Nombre de lancers de pièce » dans le rond qui suit le mot « répéter ». J'attache ce bloc au bloc précédent. En utilisant le bloc de contrôle « Répéter », je m'assure que la pièce de monnaie sera lancée autant de fois que l'utilisatrice ou l'utilisateur le souhaite.

7. Dans mon bloc de contrôle « Répéter », j'insère une variable « Mettre pièce de monnaie » à laquelle j'ajoute un bloc d'opérateur « nombre aléatoire entre 0 et 1 ». 0 représente « Pile » et 1 représente « Face ».

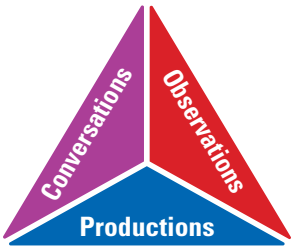
8. J'utilise le bloc « si <> alors, sinon », ce qui signifie que SI le nombre choisi est 0, ce sera pile, ALORS il s'ajoute à la liste « Pile ». SINON, ce sera face, donc il s'ajoute à la liste « Face ».

PARTIE 2 – PRATIQUE AUTONOME

Déroulement

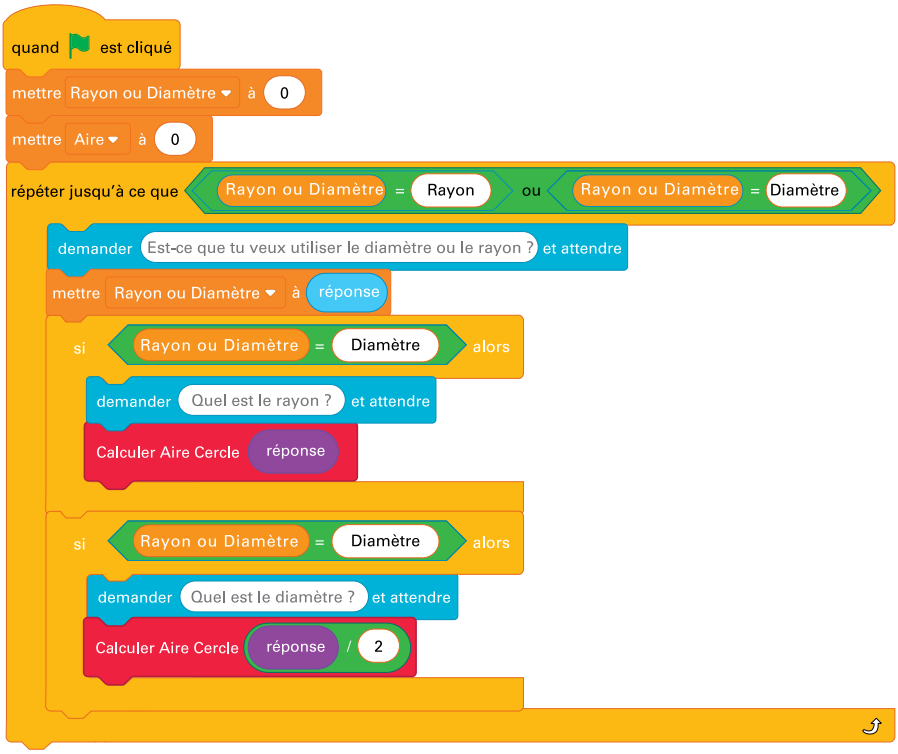
- Au besoin, demander aux élèves de faire quelques exercices de la section **À ton tour!**. Ces exercices peuvent servir de billet de sortie ou autre.
- Recueillir les preuves d'apprentissage des élèves et les interpréter pour déterminer leurs points forts et cibler les prochaines étapes en vue de les aider à s'améliorer.

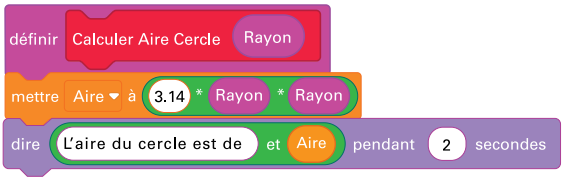
  Note : Consulter le corrigé de la partie 2, s'il y a lieu.



CORRIGÉ

1. Crée un code qui calcule l'aire d'un cercle en utilisant le diamètre ou le rayon. Identifie les blocs qui correspondent au dénombrement prédéfini ainsi qu'au sous-programme et explique leur fonction.





Je sais que pour calculer l'aire du cercle, je peux utiliser soit le rayon soit le diamètre du cercle. Je sais que l'aire du cercle est égale à pi x rayon x rayon. Je sais aussi que le diamètre peut être divisé par deux pour obtenir le rayon. Je dois donc créer un code qui va utiliser le rayon ou le diamètre pour calculer l'aire.

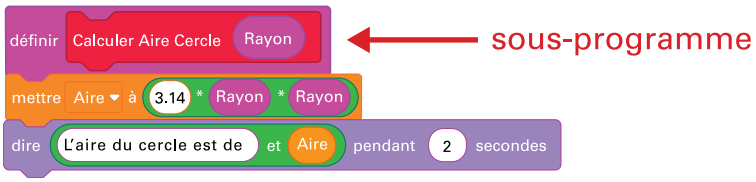
1. Je débute ma séquence avec le bloc de contrôle « Quand le drapeau vert est cliqué ».
2. Je crée deux variables qui se nomment « Rayon ou diamètre » et « Aire ». Je réinitialise toutes les informations qui sont présentes dans les variables.
3. J'utilise le bloc de contrôle « Répéter jusqu'à ce que » et j'y insère un bloc d'opérateur, afin que l'utilisatrice ou l'utilisateur puisse utiliser le rayon ou le diamètre pour obtenir l'aire d'un cercle.
4. Dans mon bloc « Répéter », j'insère le bloc de capteur « Demander » pour savoir si l'utilisatrice ou l'utilisateur désire utiliser le rayon ou le diamètre pour obtenir l'aire d'un cercle.
5. Je crée une variable afin que l'utilisatrice ou l'utilisateur puisse choisir sa réponse, soit le rayon ou le diamètre.

6. J'ajoute un bloc de contrôle « Si ». Si l'utilisatrice ou l'utilisateur choisit le rayon, elle ou il doit entrer la mesure du rayon pour obtenir l'aire du cercle. Si l'utilisatrice ou l'utilisateur choisit le diamètre, elle ou il doit entrer la mesure du diamètre pour obtenir l'aire du cercle. Sous la catégorie « Mes blocs », je crée un nouveau bloc que je nomme « Calculer Aire Cercle » avec une entrée que je nomme « Rayon ». Ceci est mon sous-programme qui me permet de calculer l'aire.

7. Sous la catégorie « Variables », j'utilise le bloc « mettre Aire à ». J'ajoute un bloc d'opérateur de multiplication. J'indique pi dans le premier rond (3.14) et dans le deuxième rond, j'insère un bloc d'opérateur de multiplication. Dans les ronds de ce dernier, j'insère les variables « rayon » et « rayon ».

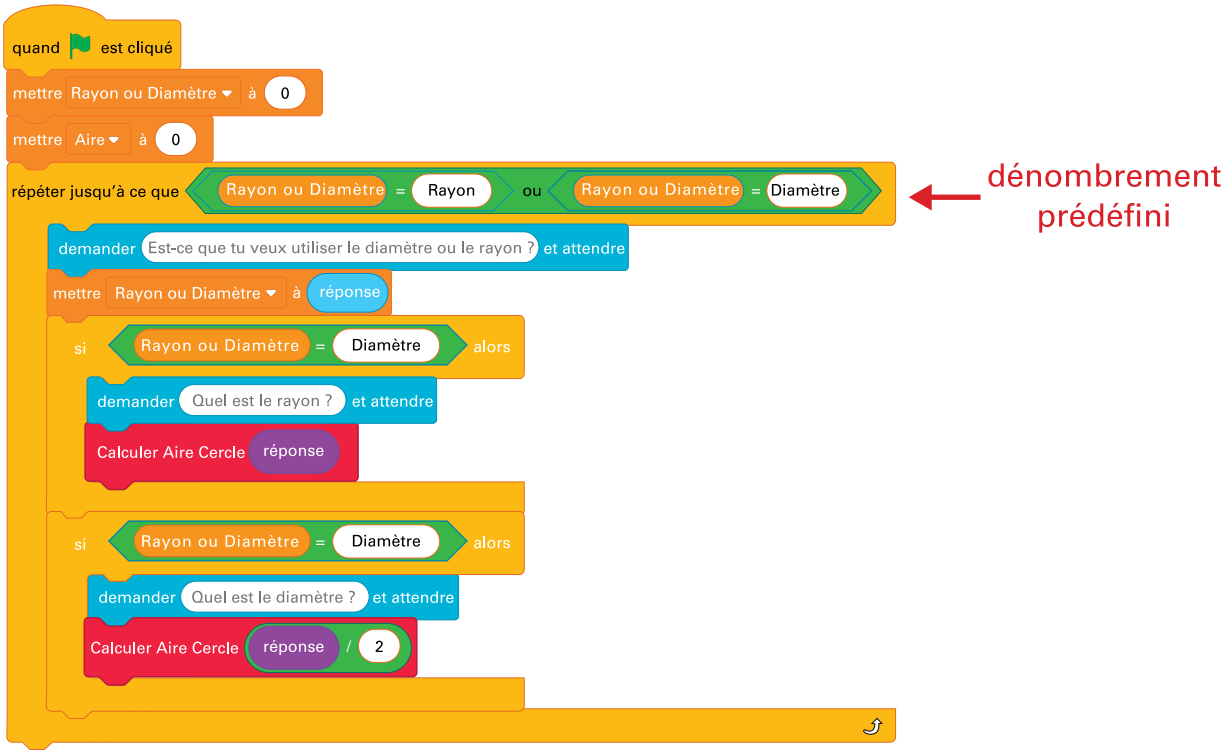
8. J'utilise le bloc d'apparence « Dire » afin de montrer l'aire du cercle. J'insère l'opérateur « regrouper … et … ». Dans le premier rond, j'écris « L'aire du cercle est de » et dans le deuxième rond, j'insère la variable « Aire ».

   Le sous-programme est le bloc rose : Définir Calculer Aire Cercle, Rayon.



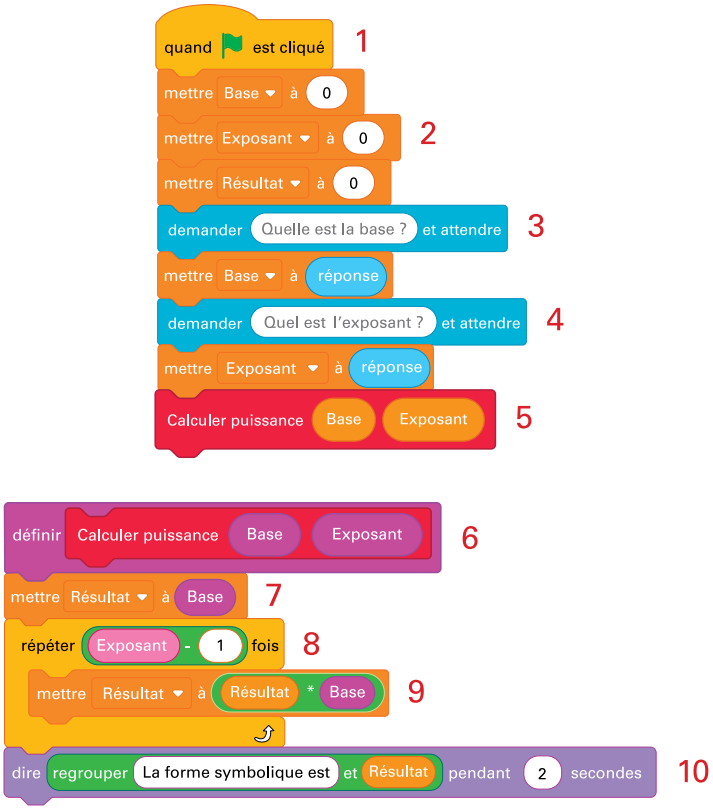
   Le sous-programme permet d'effectuer une tâche simple. Dans ce cas-ci, il est combiné au programme principal pour accomplir une tâche importante, soit calculer l'aire d'un cercle à partir du rayon ou du diamètre.

   Le dénombrement prédéfini est le bloc orange « Répéter jusqu'à ce que » :



Le dénombrement prédéfini permet d'introduire, ici, le nombre de fois que des instructions sont répétées jusqu'à ce que la condition soit remplie, soit définir l'aire d'un cercle à partir du diamètre ou du rayon.

2. Crée un code qui calcule la puissance d'un nombre à l'aide du dénombrement prédéfini et d'un sous-programme.

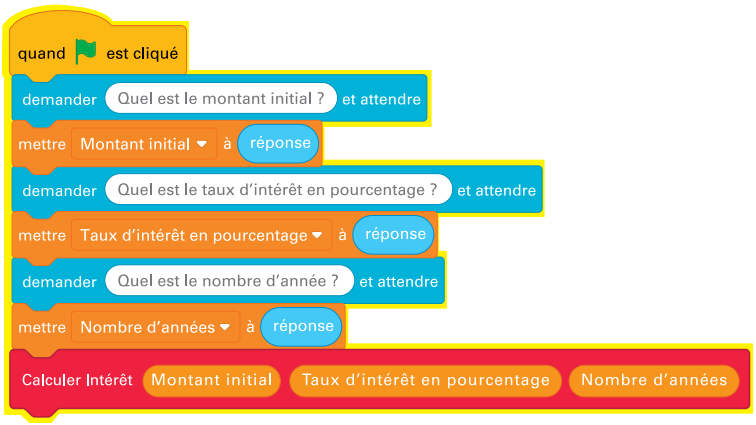


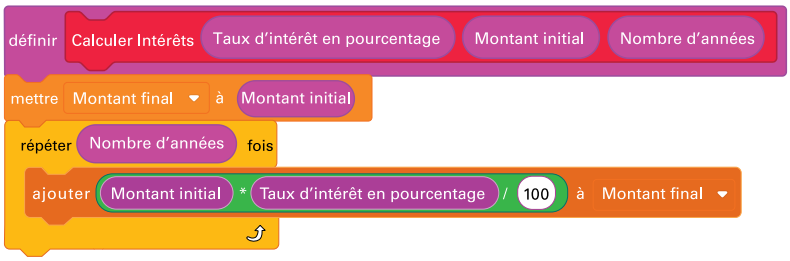
Je sais que pour calculer la puissance d'un nombre, je dois m'appuyer sur l'exposant qui fait référence au nombre de fois que je dois multiplier la base par elle-même. Je multiplie ensuite la base par elle-même pour obtenir la puissance du nombre donné.

1. Je débute ma séquence avec le bloc de contrôle « Quand le drapeau vert est cliqué ».
2. Je crée trois variables qui se nomment : Base, Exposant, Résultat.
   Je réinitialise toutes les informations qui sont présentes dans les variables.
3. J'utilise le bloc de capteur « Demander » afin de connaître la base. Je place ce nombre entré par l'utilisatrice ou l'utilisateur dans la variable « Base ».
4. J'utilise le bloc de capteur « Demander » afin de connaître l'exposant. Je place ce nombre entré par l'utilisatrice ou l'utilisateur dans la variable « Exposant ».
5. Sous la catégorie « Mes blocs », je crée un bloc que je nomme « Calculer puissance » auquel j'ajoute une entrée « Base » et une entrée « Exposant ». Ceci est le début de mon sous-programme.

6. Je définis le sous-programme « Calculer puissance ».
7. J'ajoute un bloc de la catégorie « Variables » (mettre résultat à …). Je glisse la variable « Base » dans le rond. Je mets donc le résultat égal à la base.
8. J'ajoute le bloc de contrôle « répéter … fois » en insérant un bloc d'opérateur de soustraction dans le rond. Je glisse le variable « Exposant » dans le premier rond de la soustraction et je tape 1 dans le deuxième rond de la soustraction.
9. Dans la première itération de la boucle, je multiplie la valeur de la base que j'ai mise dans la variable « Résultat » avant la boucle par la valeur de la base. La valeur de la variable « Résultat » devient le résultat de cette première multiplication. Ensuite, lors de la deuxième itération de la boucle, je multiplie le résultat de la première multiplication par la base. Je répète jusqu'à ce que toutes les multiplications désirées soient effectuées.
10. J'utilise le bloc apparence « Dire » afin de montrer la forme symbolique de la puissance.

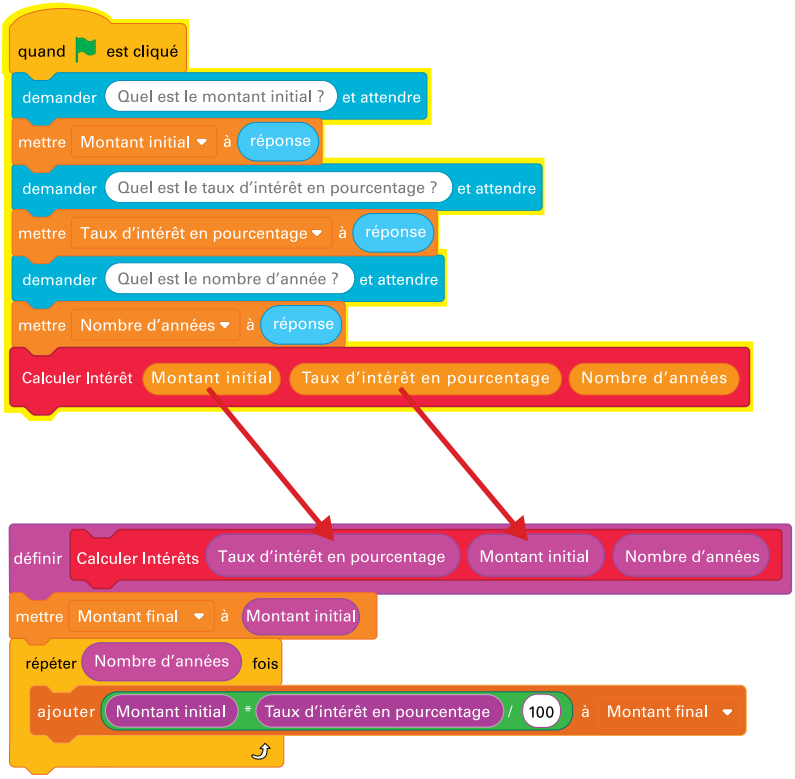
3. Une erreur s'est glissée dans ce code qui calcule le taux d'intérêt sur un montant d'argent donné pendant une période donnée. Peux-tu la trouver?



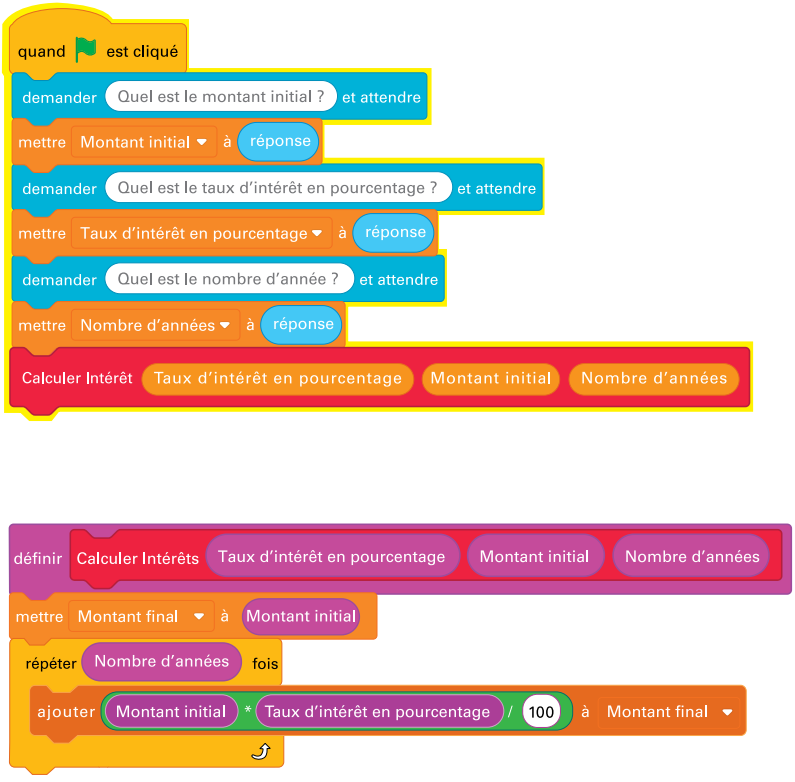


En exécutant le code, je remarque que si je mets un montant initial de 1 000,00 $, par exemple, et que le taux d'intérêt est de 10 % pendant une durée de deux ans, le montant final s'élève à 210,00 $, ce qui ne peut être possible. Le montant final doit être supérieur au montant initial.

L'erreur se situe dans la définition du sous-programme. L'ordre établi dans le programme du haut doit est le même que dans le sous-programme.



Je modifie donc le code en conséquence et j'exécute la séquence à nouveau :



4. Voici des blocs qui, une fois assemblés, servent à trouver les facteurs d'un nombre.

a) Peux-tu les placer en ordre afin d'en faire un code efficace?

Solution

1. Je débute ma séquence avec le bloc de contrôle « Quand le drapeau vert est cliqué ».
2. Sous la catégorie « Variables », je crée une liste que je nomme « Liste Facteurs » et je réinitialise toutes les informations qui sont présentes dans la liste.
3. J'utilise le bloc de capteur « Demander » afin de savoir le nombre pour lequel je veux trouver les facteurs.
4. Sous la catégorie « Mes blocs », je crée un bloc que je nomme « Trouver Facteurs » et j'y ajoute une entrée dans laquelle je glisse le capteur « réponse ».
5. Je définis le sous-programme « Trouver Facteurs ».
6. Je donne à la variable « Facteurs Possibles » la valeur du nombre entré par l'utilisatrice ou l'utilisateur. Cette variable descend de -1 à chaque itération de la boucle afin de vérifier si le nombre suivant est, lui aussi, un facteur.
7. En utilisant le bloc de contrôle « Répéter », je m'assure que l'utilisatrice ou l'utilisateur peut entrer le nombre de données souhaité.

8. J'utilise le bloc de contrôle « si < > alors ». Ensuite, j'insère le bloc modulo. En informatique, le modulo est en quelque sorte utilisé pour connaître le reste d'une division. Donc, si un nombre se divise par un autre nombre sans reste, le modulo sera 0. Dans ce bloc, ce sont les nombres qui ont un modulo de 0 qui sont recherchés, car cela signifie qu'ils sont facteurs.

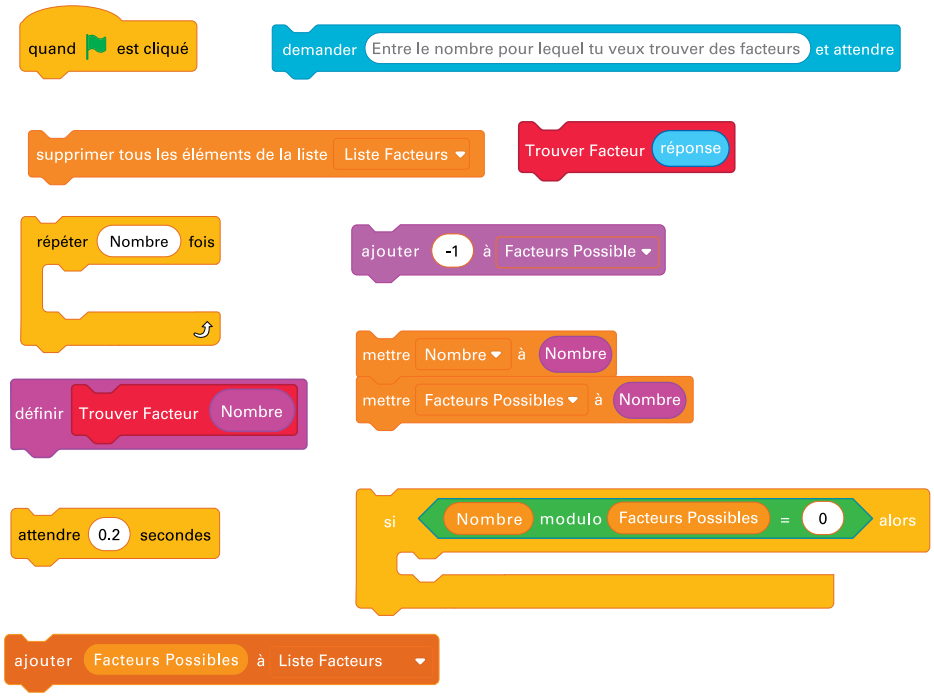
   Exemple : 12 modulo 2 va donner 0, car 2 entre exactement 6 fois dans 12 sans donner de reste.

9. Si le nombre dans la variable « Facteurs Possibles » est un facteur, ce nombre est ajouté à la liste.

10. Je place le bloc qui permet d'attendre afin de voir les facteurs s'ajouter à la liste. Ce bloc est optionnel.

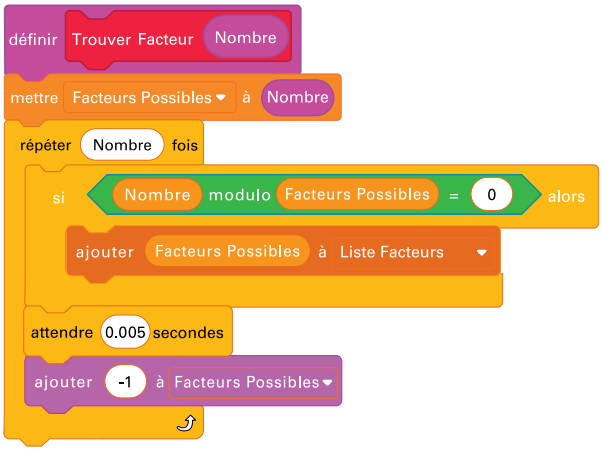
11. J'ajoute le bloc « ajouter -1 » qui permet de faire passer le prochain nombre (facteur possible) dans la boucle. Par exemple, si le nombre entré est 4, ensuite ce sera 3, puis 2 et ainsi de suite. Ici, il est important de mettre cette étape en dehors du bloc « SI » car nous voulons ajouter -1 peu importe si un facteur a été trouvé ou non durant l'itération en cours.

b) Peux-tu trouver le dénombrement prédéfini et le sous-programme?

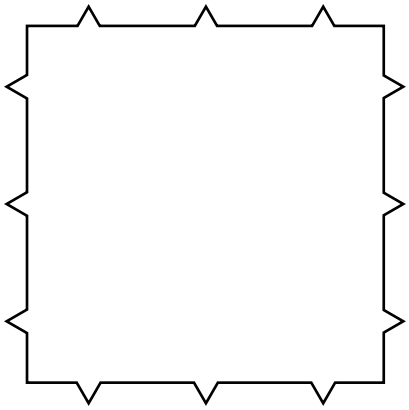


Le dénombrement prédéfini est le bloc : Répéter Nombre fois. Je le sais parce que le dénombrement prédéfini correspond au nombre de fois que des instructions sont répétées en fonction d'une valeur prédéfinie.

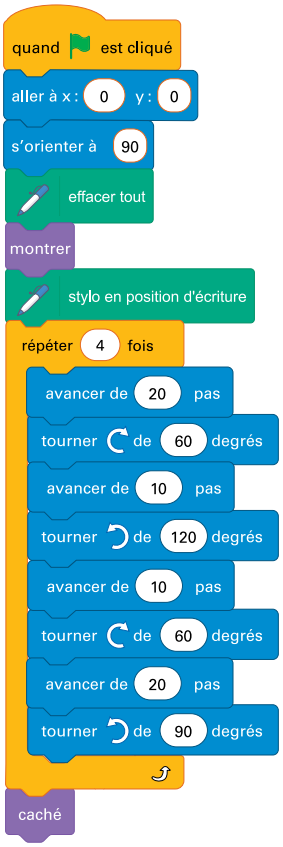
Le sous-programme est le bloc « Définir Trouver Facteurs Nombre ». Je le sais car un sous-programme correspond à un petit ensemble d'instructions qui permet d'effectuer une tâche simple, soit d'identifier les facteurs d'un nombre.



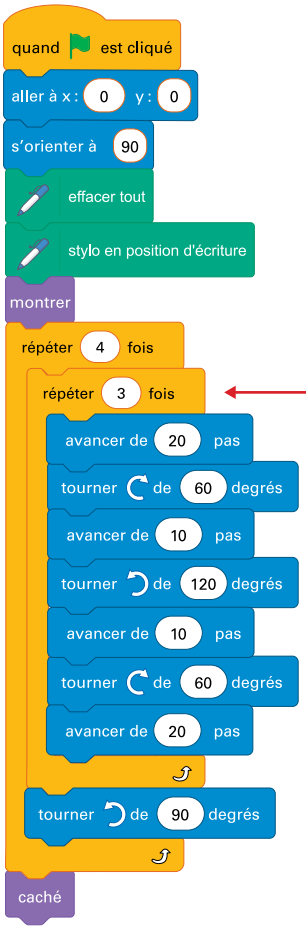
5. Julianne a créé un code de façon à obtenir le dessin suivant :



Malheureusement, en exécutant son code, elle remarque qu'elle n'obtient pas le résultat souhaité. Aide Julianne à trouver ses erreurs et modifie le code en conséquence.



En lisant le code de Julianne, je remarque qu'il manque une boucle à l'intérieur de façon à obtenir le résultat souhaité.



J'ajoute un bloc de contrôle « Répéter » afin d'obtenir une boucle

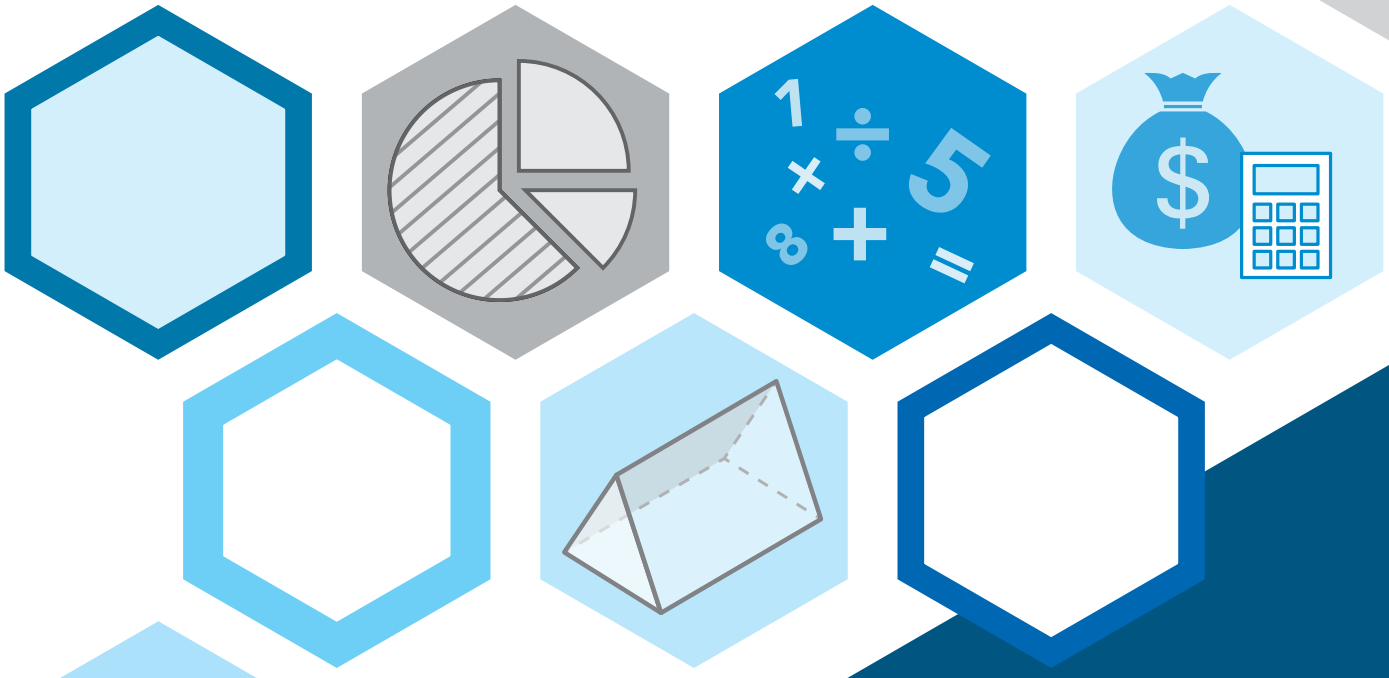
Version de l'élève

7e année

En avant, les maths!

Une approche renouvelée pour l'enseignement et l'apprentissage des mathématiques

MINILEÇON



ALGÈBRE

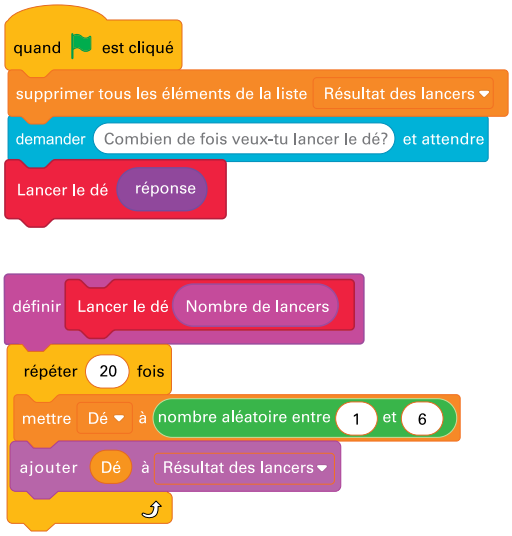
Créer et modifier des codes comprenant des événements influencés par un dénombrement prédéfini et/ou un sous-programme et d'autres structures de contrôle

PARTIE 1 – EXPLORATION GUIDÉE

EXEMPLE 1

a) À l'aide d'un logiciel de programmation de ton choix, crée un code contenant un dénombrement prédéfini et un sous-programme permettant de faire rouler un dé un certain nombre de fois.

b) Voici un code qui offre la possibilité à l'utilisatrice ou à l'utilisateur de déterminer le nombre de fois qu'elle ou qu'il souhaite lancer un dé à 20 faces. Cependant, le code comporte deux erreurs. Peux-tu les identifier? Modifie le code et décris l'incidence de ces changements sur les résultats et l'efficacité.



TA STRATÉGIE

EXEMPLE 2

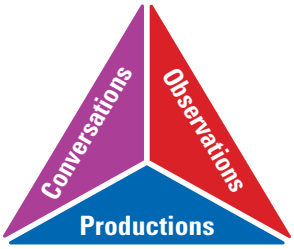
a) À l'aide d'un logiciel de programmation de ton choix, crée un code qui fait tourner une pièce de monnaie 10 fois à l'aide d'un dénombrement prédéfini.

b) Modifie ton code de façon à ce que l'utilisatrice ou l'utilisateur puisse choisir le nombre de fois qu'elle ou il désire faire tourner la pièce de monnaie en y insérant un sous-programme.

TA STRATÉGIE

PARTIE 2 – PRATIQUE AUTONOME

À ton tour!



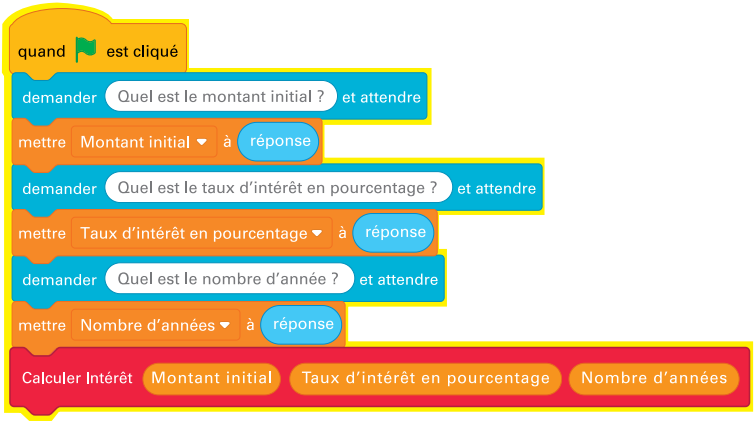
1. Crée un code qui calcule l'aire d'un cercle en utilisant le diamètre ou le rayon. Identifie les blocs qui correspondent au dénombrement prédéfini ainsi qu'au sous-programme et explique leur fonction.

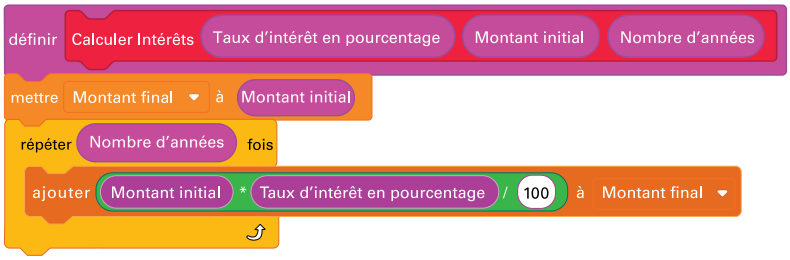
TA STRATÉGIE

2. Crée un code qui calcule la puissance d'un nombre à l'aide du dénombrement prédéfini et d'un sous-programme.

TA STRATÉGIE

3. Une erreur s'est glissée dans ce code qui calcule le taux d'intérêt sur un montant d'argent donné pendant une période donnée. Peux-tu la trouver?



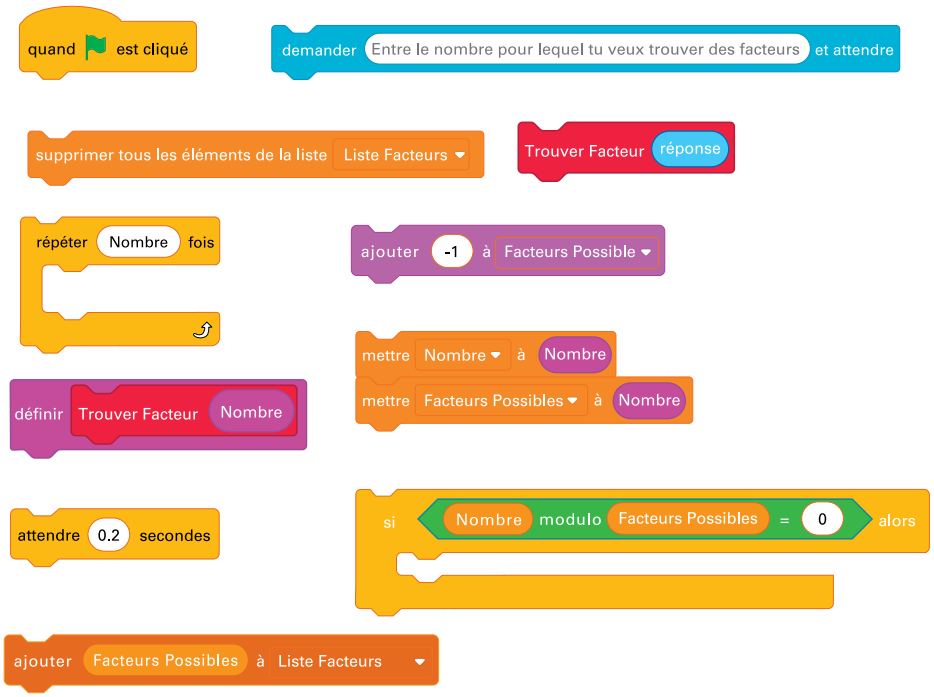


TA STRATÉGIE

4. Voici des blocs qui, une fois assemblés, servent à trouver les facteurs d'un nombre.

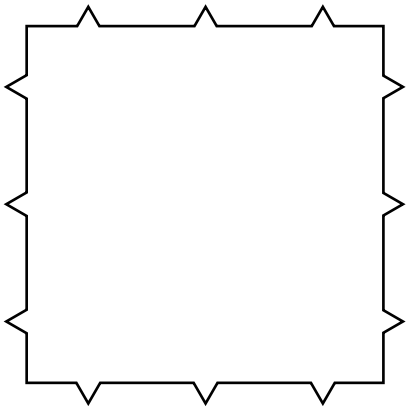
   a) Peux-tu les placer en ordre afin d'en faire un code efficace?

   b) Peux-tu trouver le dénombrement prédéfini et le sous-programme?

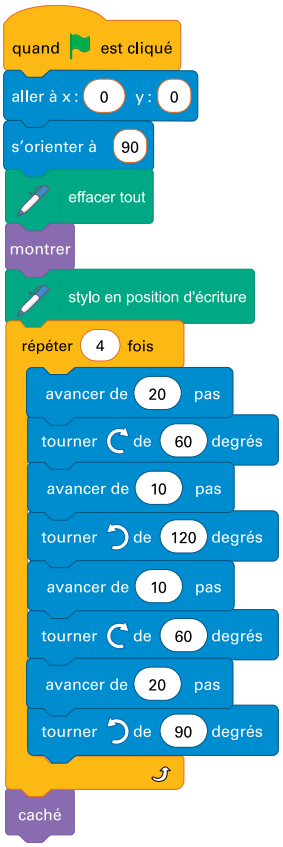


TA STRATÉGIE

5. Julianne a créé un code de façon à obtenir le dessin suivant :



Malheureusement, en exécutant son code, elle remarque qu'elle n'obtient pas le résultat souhaité. Aide Julianne à trouver ses erreurs et modifie le code en conséquence.



TA STRATÉGIE