

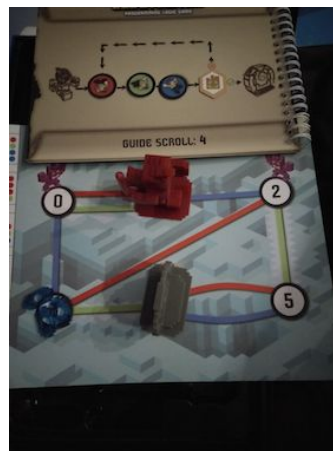


## Étape 5 - Les Boucles

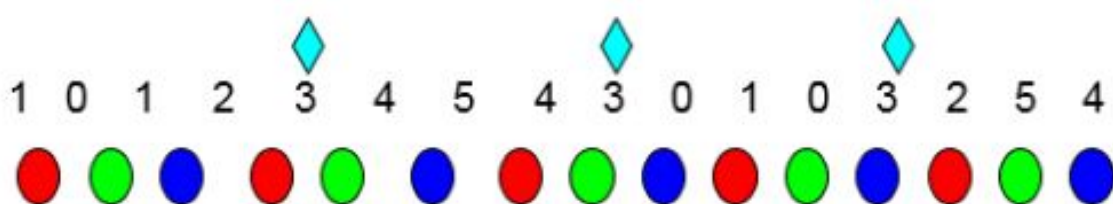
### Code Master

#### Notes

Reprenons le niveau 33 (Map 3 - Scroll 4). La solution est la suivante (rouge-vert-bleu) :



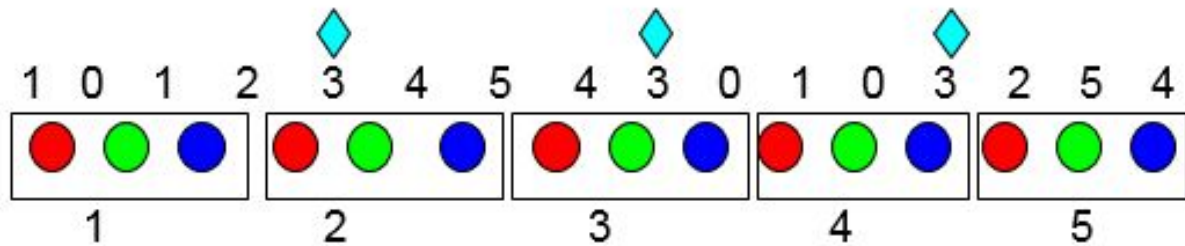
Déroulons le programme :



Nous pouvons voir que les instructions rouge - vert - bleu sont répétées jusqu'à ce que la condition de sortie soit **VRAIE**. Dans ce cas, la condition de sortie est :

`NombreDiamants est égal à 3 ET il y a un Troll orange dans la case courante.`

## Premiers pas en Algorithmie



Dans l'exemple, les instructions sont répétées 5 fois :

Et nous pouvons regrouper ces instructions dans une **boucle** :

**TantQue condition est FAUX**



**Fin TantQue**

Dans cette boucle, les instructions rouge - vert -bleu sont répétées **tant que** la condition de sortie, décrite précédemment, n'est pas **FAUSSE**. Les boucles sont très importantes en informatique, car elle permet (entre autres) de :

- Réduire les lignes de code (on évite d'écrire du code redondant)
- Exécuter des instructions tant qu'une condition n'est pas **VRAIE**
- Le nombre des fois que les instructions sont répétées peut dépendre des entrées et les valeurs des variables de notre algorithme.

Grâce aux boucles, nous obtenons un code plus compact et plus générique.

### Modalités

- Travail en équipe (4 personnes)
- Production individuelle

### Objectifs

- Comprendre, à l'aide de "Code Master", les boucles en programmation
- Déroulez les niveaux 51 à 60. Formalisez sur papier les boucles et les conditions nécessaires pour sortir de ces boucles.
- Ecrivez au tableau le pseudo-code du niveau 54.

## Pseudo-code + Blockly

### Énoncé 1

Écrire un programme qui

- Initialise une variable « n » à 1
- multiplie « n » par 2 **tant que** « n » est inférieur à 10
- Afficher « n »

### Énoncé 2

Écrire un programme qui

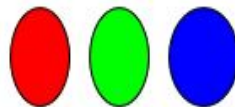
- sélectionne un nombre aléatoirement entre 1 et 100 et le met dans une variable appelée « secret »
- vous demande une valeur « n » **tant que** « n » est différent de « secret »
  - quand « secret » est plus grand que “n” afficher “plus grand”
  - quand « secret » est plus petit que “n” afficher “plus petit”

## Les boucles « Pour »

Il y a différents types de boucles : les boucles **TantQue** et les boucles **Pour**.

Les boucles **Pour** nous permettent de répéter les instructions un nombre déterminé de fois. Dans notre exemple, si nous voulons répéter les instructions rouge - vert - bleu cinq fois nous pouvons faire :

Variable Count: Entier  
Pour Count de 1 à 5



## Sur papier

Convertissez la boucle **TantQue** suivante en une boucle **Pour**

```
Variable X : Entier
Variable Total : Entier
Debut
X ← 0
Total ← 0
TantQue X < 10
  X ← X + 1
  Total ← Total + X
```

**Fin TantQue**

**Fin**

**Question :**

Quelle est la valeur finale de Total ?

**Ressources**

- <http://pise.info/algo/boucles.htm>
- <http://www.charlie-soft.com/Programmation/Algorithmique/Boucles.php>

## Pseudo-code + Blockly

**Énoncé**

- Écrire un programme qui calcule la somme des nombres de 1 à 100 inclus.

## Étape 6 - Les Tableaux

Un tableau est une structure de données utilisée en programmation. Elle consiste en un ensemble de données du même type (texte, entier, décimal ...)

Voyons un exemple du tableau :

```
1- Tableau Nombres(7) en Entier  
2-   Nombres ← [1, 2, 5, 8, 5, 7, 8]
```

Dans la ligne 1, nous déclarons la variable **Nombres** qui est un tableau de 7 valeurs.

Dans la ligne 2, nous affectons la variable avec la liste de valeurs (1,2,5,8,5,7,8).

Pour accéder aux valeurs de notre tableau, nous utilisons un **index**.

```
3- Nombres[0]  
4- Nombres[6]
```

Dans la ligne 3, nous accédons à la première valeur de notre tableau : 1

Dans la ligne 4, nous accédons à la dernière valeur de notre tableau : 8

Les tableaux sont très utilisés dans les boucles, par exemple pour accéder à toutes les valeurs de notre tableau nous pouvons utiliser une boucle **Pour** :

```
1- Tableau Nombres(7) en Entier  
2- Variable i en Entier  
3- Début  
4- Nombres ← [1, 2, 5, 8, 5, 7, 8]  
4- Pour i de 0 à 6  
5- imprimer Nombres[i]  
6- Fin
```

## Pseudo-code + Blockly

### Modalités

- Production individuelle

### Énoncé 1

Dans l'environnement BLOCKLY

- créez un tableau de nombres entiers
- écrivez une fonction, en utilisant une boucle, qui calcule la moyenne des nombres du tableau

### Énoncé 2

Dans l'environnement BLOCKLY

- Créez un tableau vide
- Remplissez le avec 3 valeurs aléatoires entre 1 et 100 (avec une boucle)
- Afficher les 3 valeurs (avec une boucle)

### Énoncé 3

- Continuer l'exercice précédent pour afficher ensuite la plus petite valeur du tableau.
- Faire le logigramme de recherche de la plus petite valeur

### Ressources et plus d'informations :

- <http://pise.info/algo/tableaux.htm>
- <http://www.charlie-soft.com/Programmation/Algorithmique/Tableaux.php>

## Pour aller plus loin - Boutique multi-produits

Votre boutique en ligne marche bien et votre produit star se vend comme des petits pains. Vous décidez d'augmenter votre catalogue et d'ajouter plusieurs produits.

### Énoncé

- Implémentez votre liste de produits sous la forme de 3 tableaux contenant :
  - le nom de chaque produit,
  - le prix de chaque produit
  - le stock de chaque produit
- Pour chaque produit,
  - demandez à l'utilisateur la quantité demandée,
  - validez que le stock est suffisant
- À la fin de la commande, affichez le total à payer