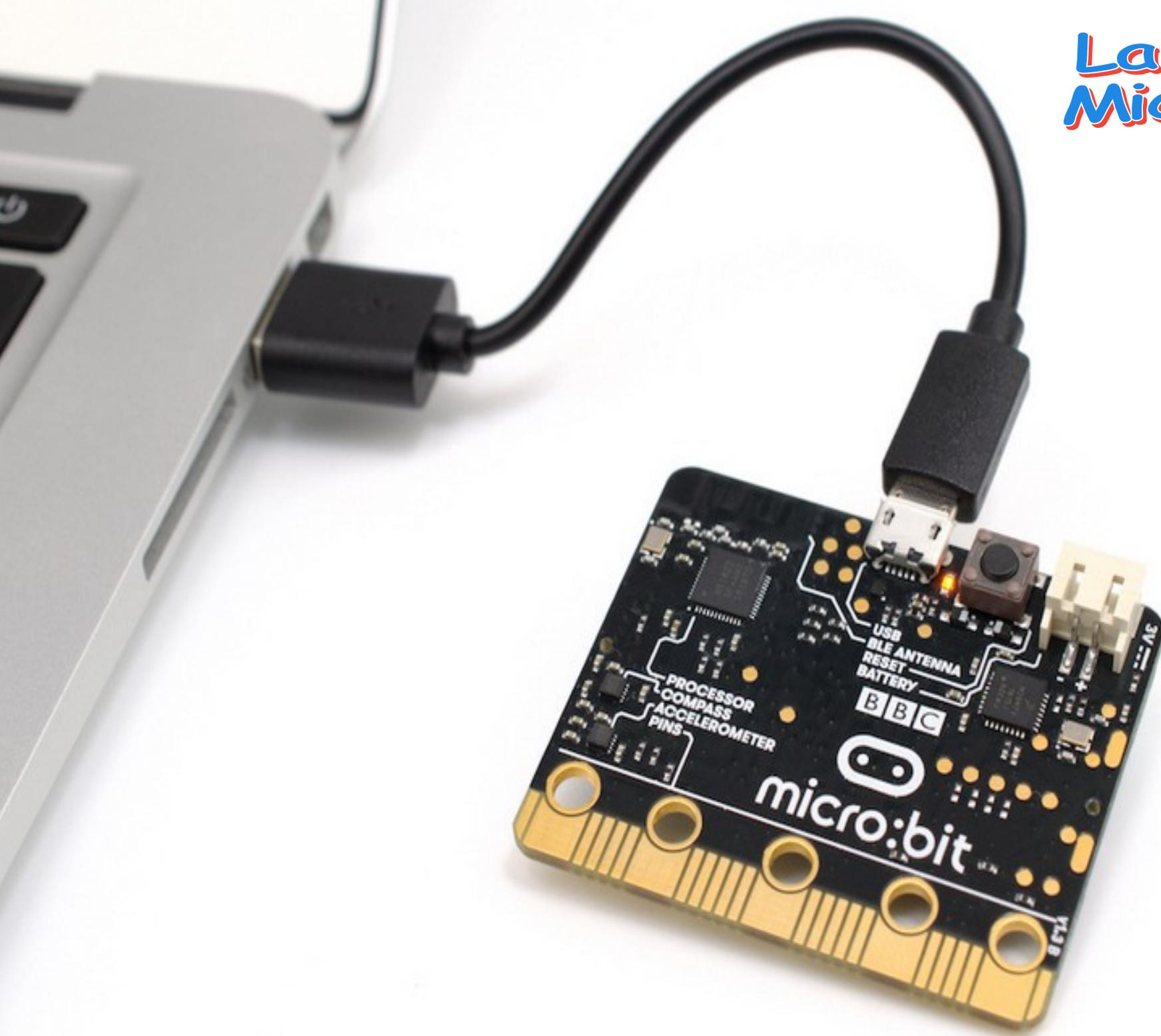


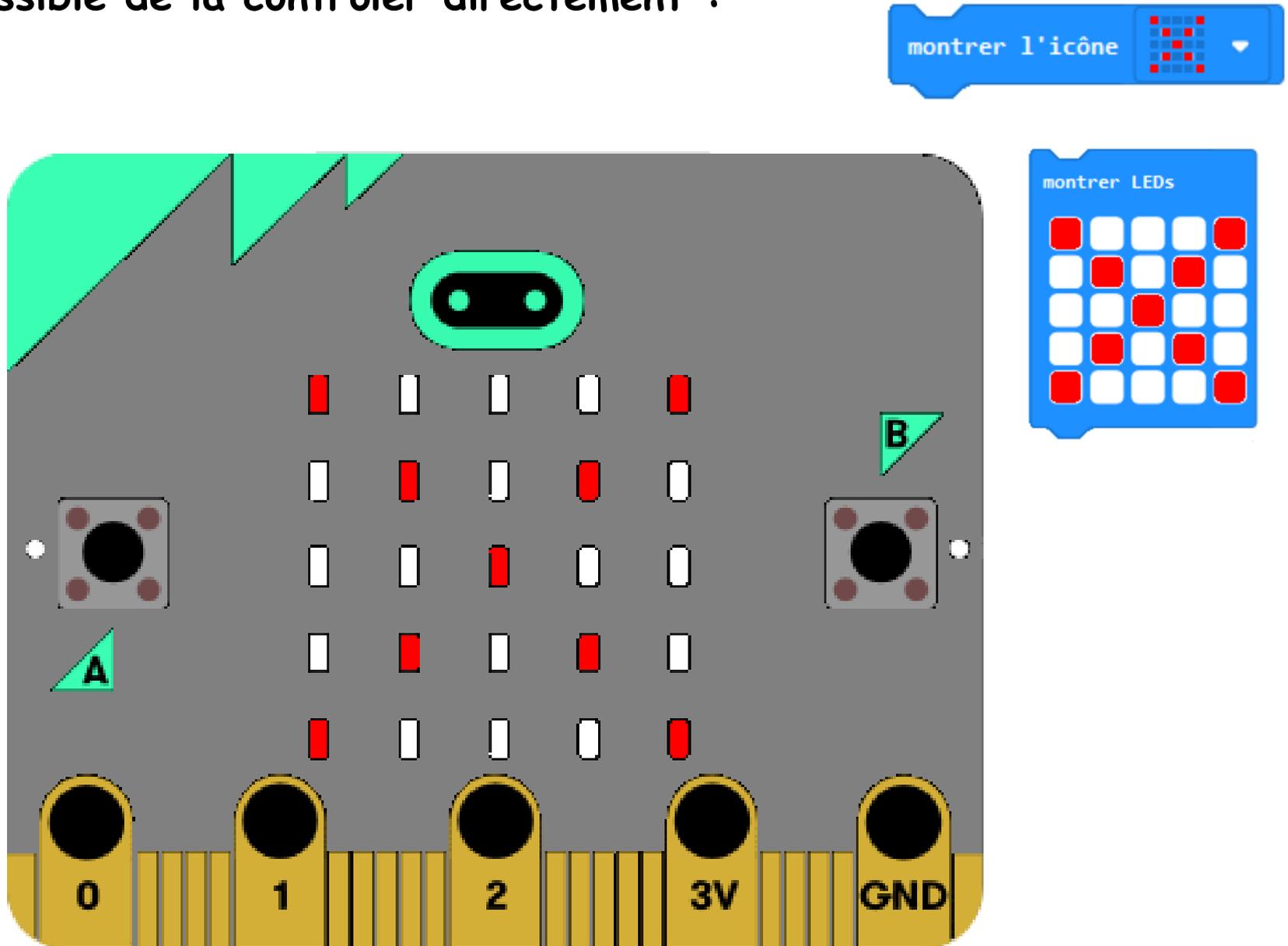
# La carte Micro:bit 2



# L'afficheur de la carte micro:bit

L'afficheur est constitué de 5 x 5 LED appelé matrice.

Il est possible de la contrôler directement :

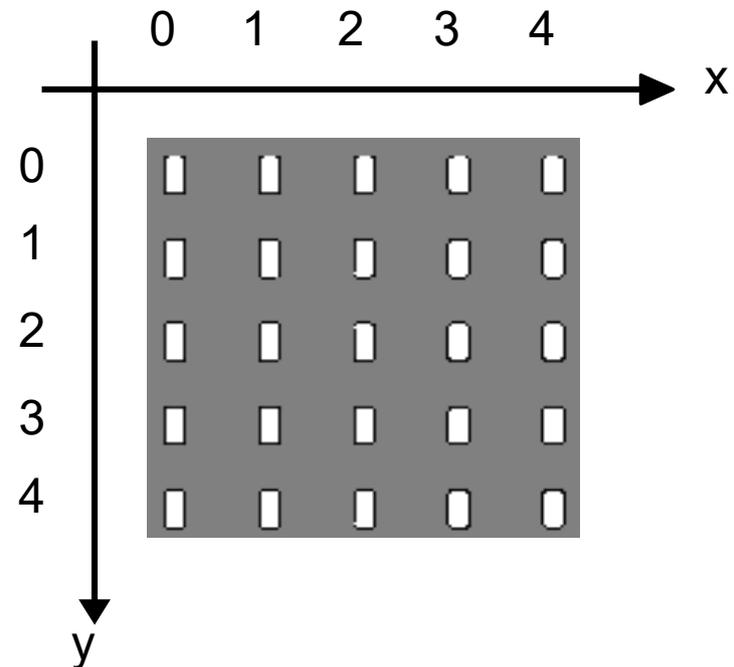
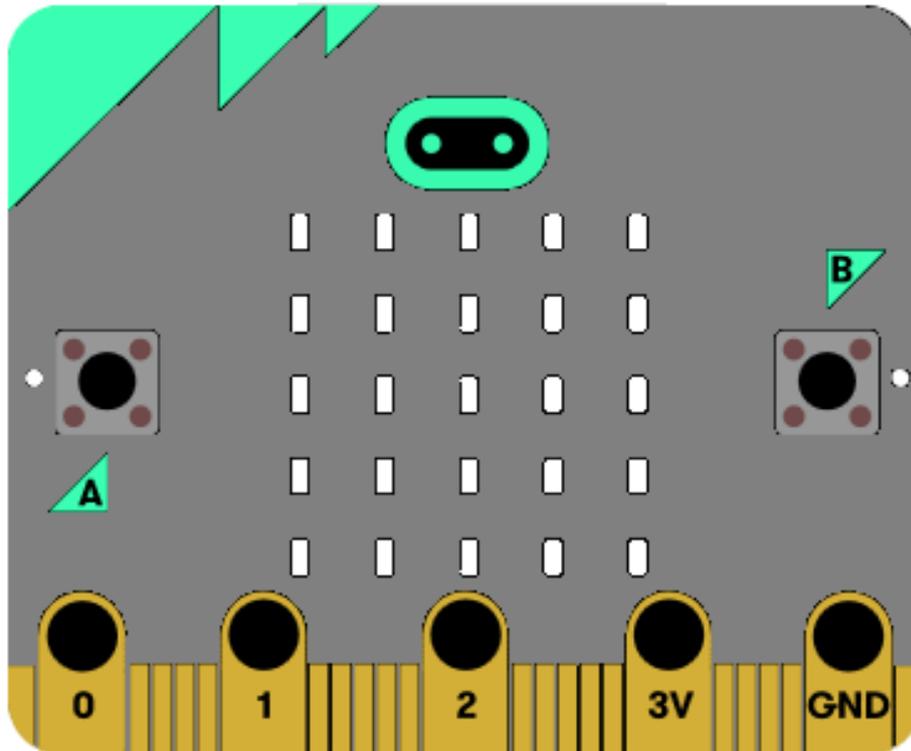


# Coordonnées de la matrice

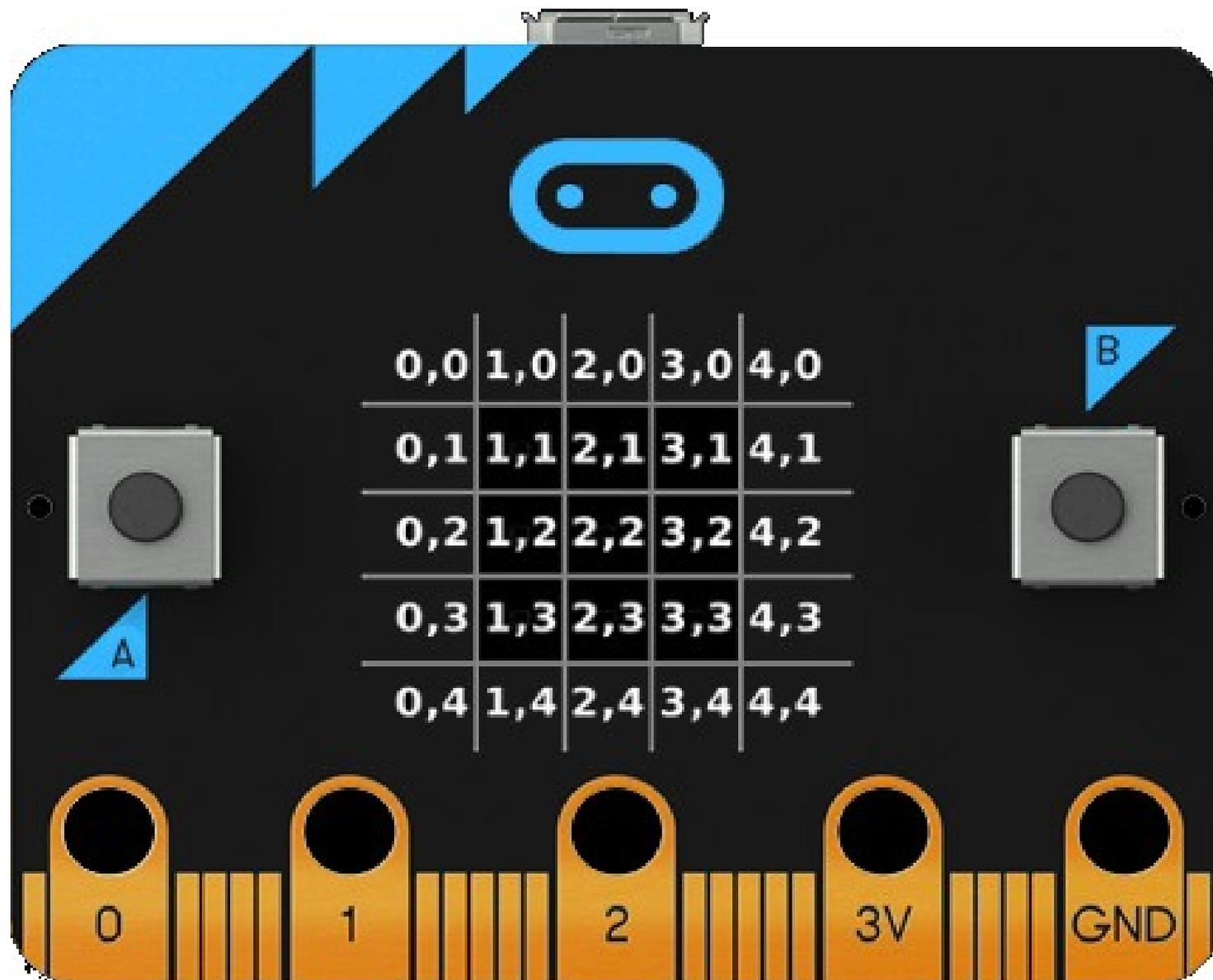
Il est possible de contrôler une seule LED dans la matrice en utilisant :

x est la position horizontale

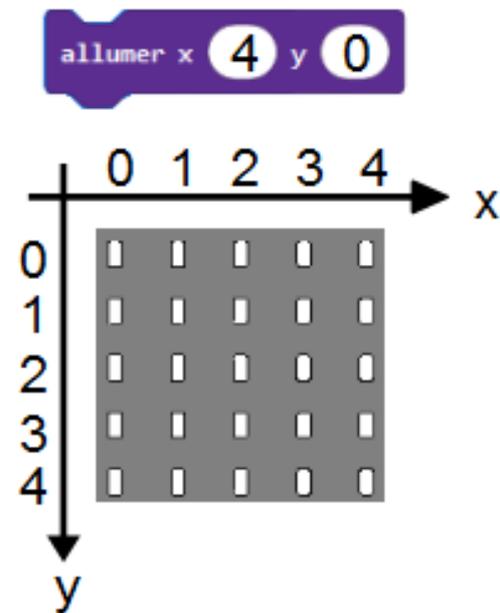
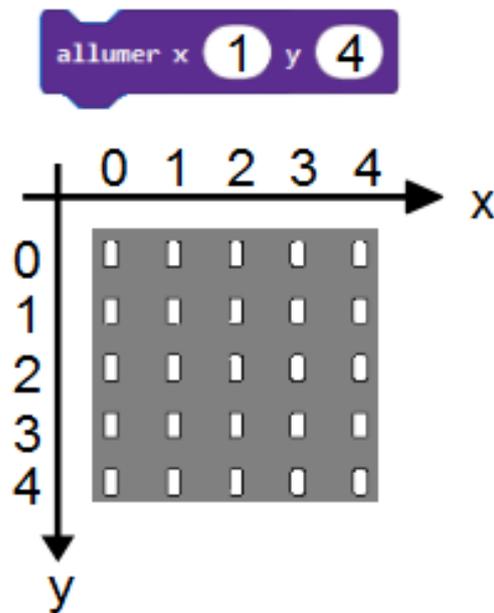
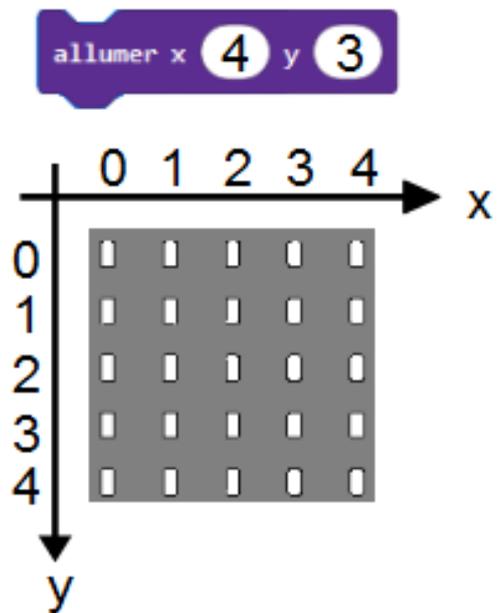
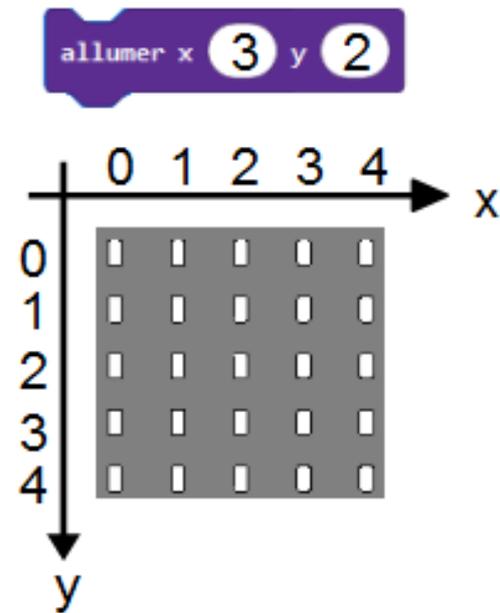
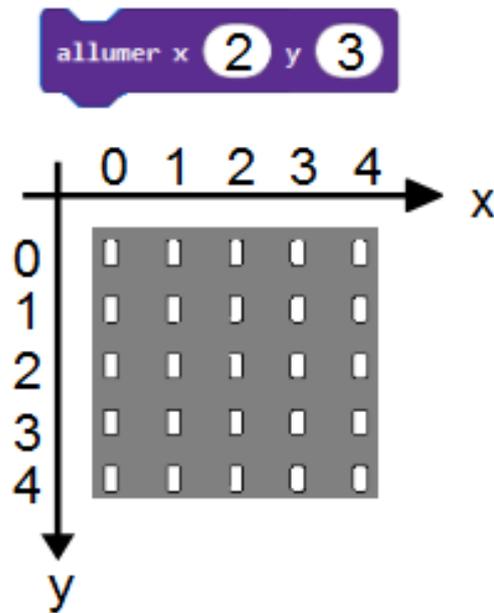
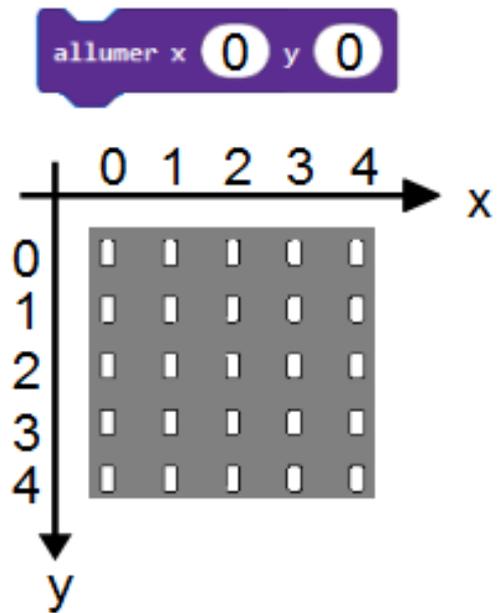
y est la position verticale



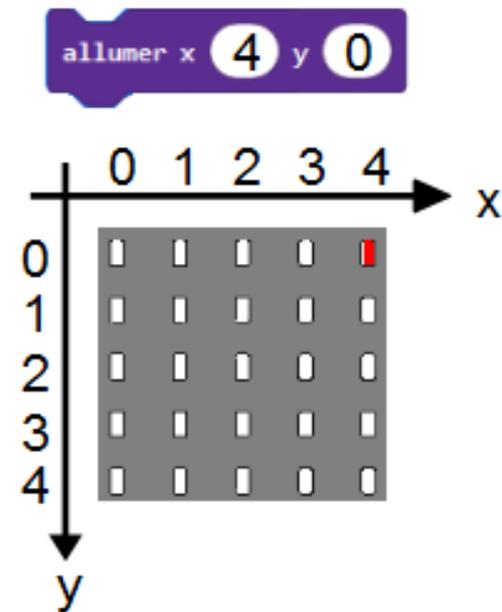
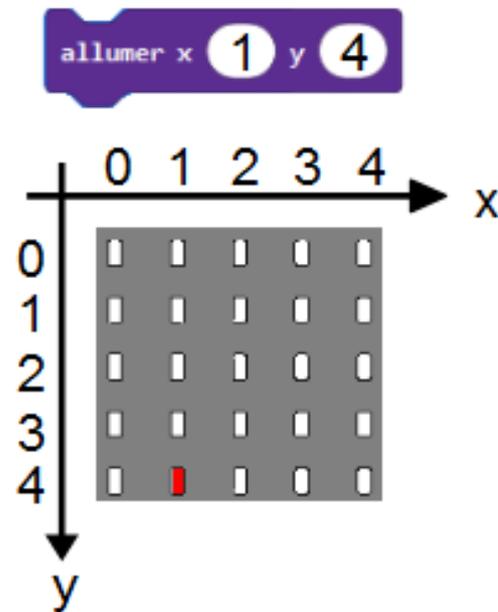
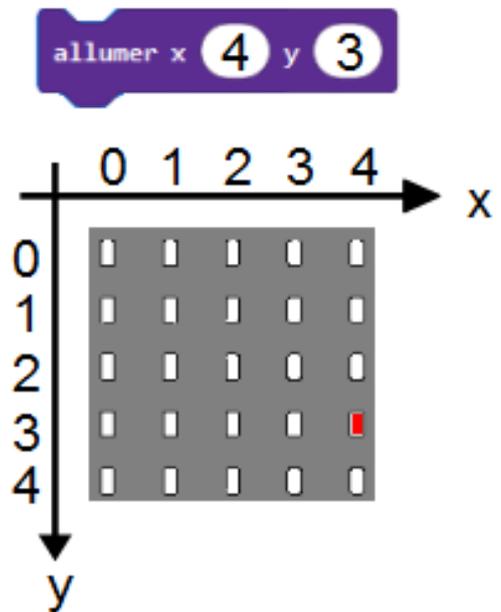
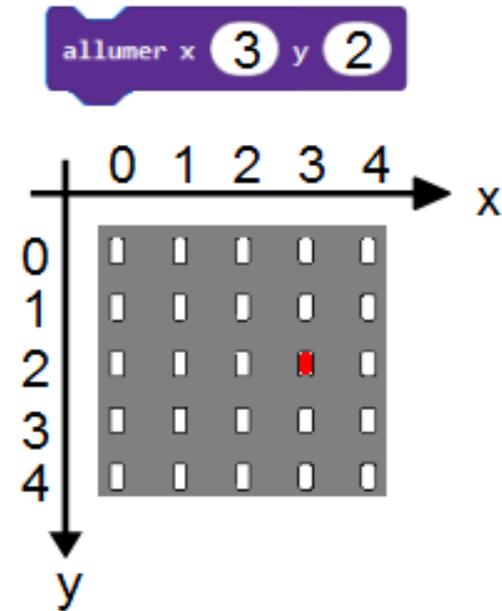
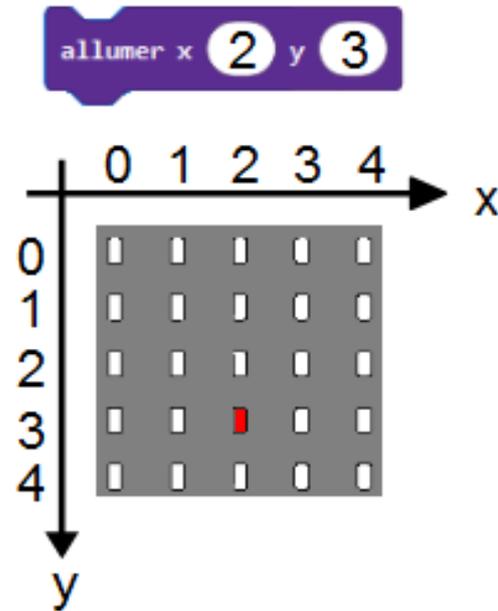
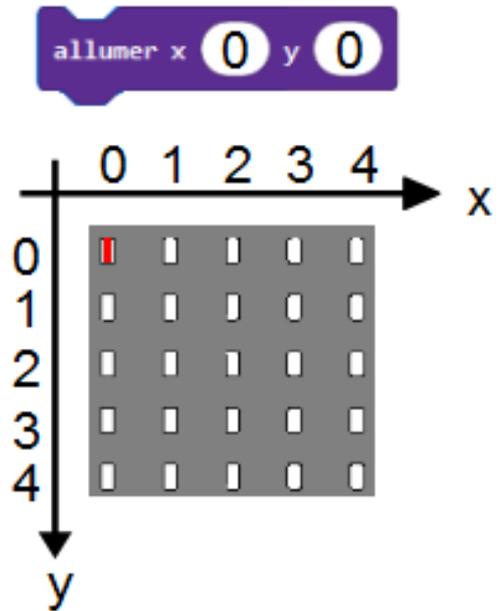
Chaque LED correspond aux coordonnées x,y



# Chaque LED correspond aux coordonnées $x,y$



# Chaque LED correspond aux coordonnées $x,y$



# Sur une feuille double :

ECRIRE :

## 1 - Gestion de l'afficheur

Il est possible de contrôler une LED en utilisant :

x est la position horizontale  
y est la position verticale

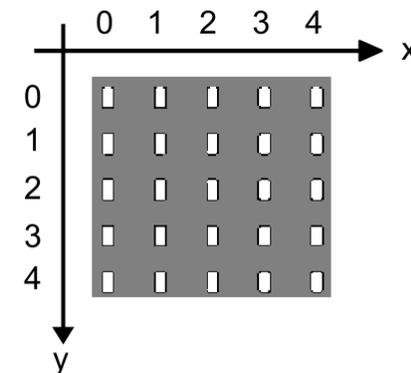
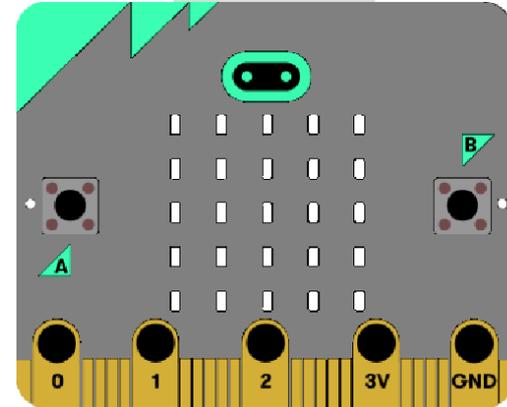
Découper et coller la carte Micro:bit ainsi que la matrice

Nom  
Prénom

Carte micro:bit Programmation2

### 1 - Gestion de l'afficheur

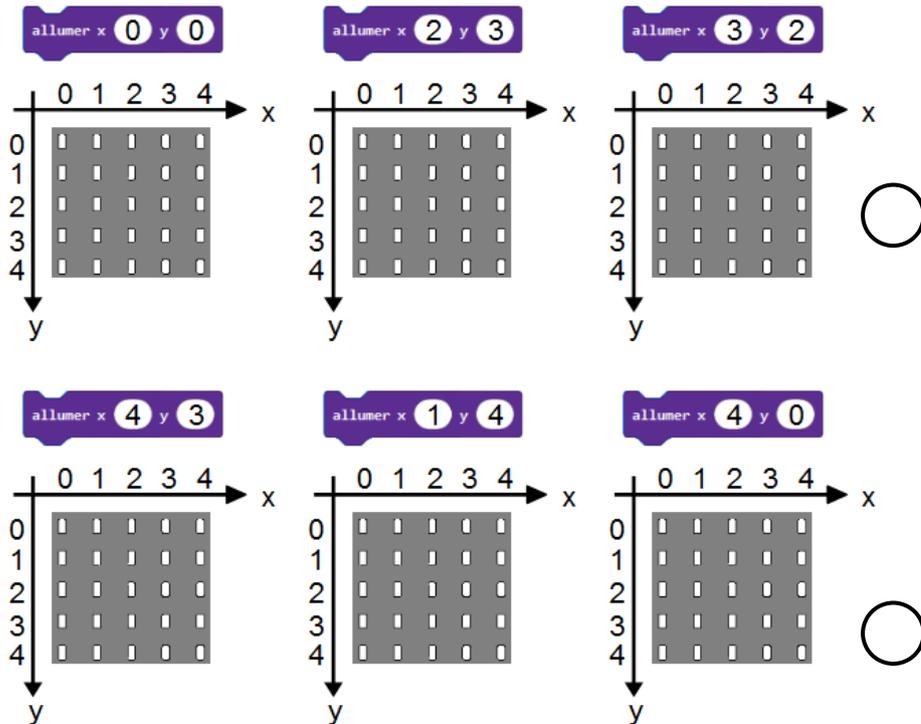
Il est possible de contrôler une LED en utilisant :  
x est la position horizontale  
y est la position verticale



# A l'intérieur de la feuille double (page 2 et 3) :

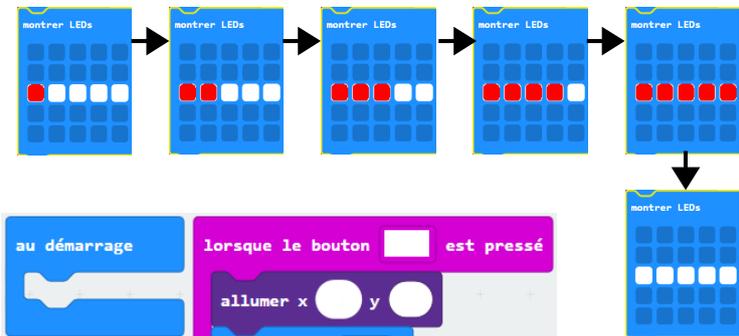
Découper et coller : les matrices ainsi que le Programme2\_1 **Compléter le programme**

Colorier en rouge la LED correspondante



## Programme2\_1 :

Quant A est pressé  
Permet de tracer une ligne au centre de la matrice puis effacer l'écran.



```
au démarrage
toujours

lorsque le bouton [ ] est pressé
  allumer x [ ] y [ ]
  pause (ms) 500
  allumer x [ ] y [ ]
  pause (ms) 500
  allumer x [ ] y [ ]
  pause (ms) 500
  allumer x [ ] y [ ]
  pause (ms) 500
  allumer x [ ] y [ ]
  pause (ms) 500
  efface l'écran
```

# Page 4 de la feuille double :

Découper et coller :

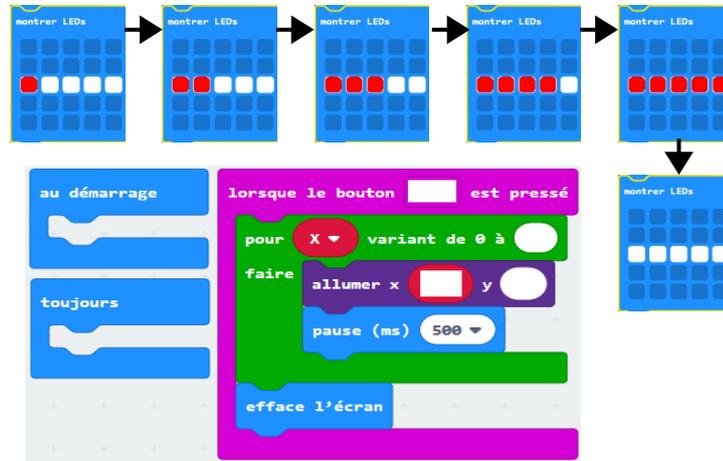
Le Programme2\_2

Le programme2\_3

Compléter les programmes  
Tester les programmes...

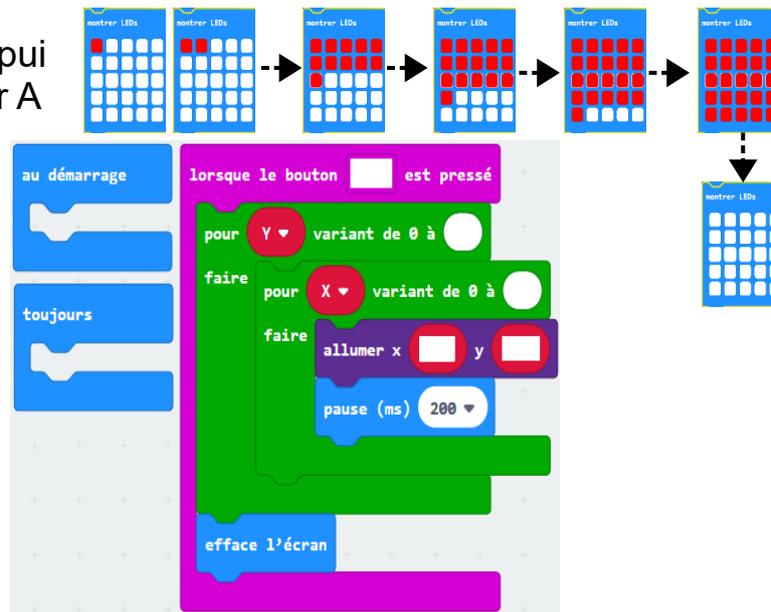
## Programme2\_2 :

Même fonctionnement que Programme2\_1  
mais utilisant la variable X



## Programme2\_3 :

Appui  
sur A



# Scotcher une feuille simple sur votre feuille double :

Découper et coller :

Le Programme2\_4

Le programme2\_5

Compléter les programmes  
Tester les programmes

The diagram illustrates the process of transferring code from a double-page program to a single-page program. It is divided into two main sections: 'Programme2\_4' and 'Programme2\_5'.

**Programme2\_4:** This section shows a sequence of four small grid icons representing the state of a program. An arrow points from the first icon to a larger code editor window. The code editor contains the following blocks:

- au démarrage** (at start):
  - Appui sur A (Push A)
  - toujours (forever loop)
- lorsque le bouton [ ] est pressé** (when button is pressed):
  - pour Y variant de 0 à [ ] (for Y from 0 to [ ]):
    - faire pour X variant de 0 à [ ] (do for X from 0 to [ ]):
      - faire allumer x [ ] y [ ] (do turn on x [ ] y [ ])
      - pause (ms) 200 (pause 200 ms)
  - pour Y variant de 0 à [ ] (for Y from 0 to [ ]):
    - faire pour X variant de 0 à [ ] (do for X from 0 to [ ]):
      - faire éteindre x [ ] y [ ] (do turn off x [ ] y [ ])
      - pause (ms) 200 (pause 200 ms)

**Programme2\_5:** This section shows a single-page code editor with the following blocks:

- au démarrage** (at start):
  - toujours (forever loop)
- lorsque le bouton [ ] est pressé** (when button is pressed):
  - répéter 2 fois (repeat 2 times):
    - faire pour Y variant de 0 à [ ] (do for Y from 0 to [ ]):
      - faire pour X variant de 0 à [ ] (do for X from 0 to [ ]):
        - faire activer/désactiver x [ ] y [ ] (do activate/deactivate x [ ] y [ ])
    - pause (ms) 200 (pause 200 ms)

Arrows indicate the transfer of code blocks from Programme2\_4 to Programme2\_5. Specifically, the 'Appui sur A' block is moved to the 'au démarrage' section, and the nested loops from the 'lorsque le bouton [ ] est pressé' section are moved to the 'répéter 2 fois' block.

# Page 6 de la feuille :

Découper et coller :

Le Programme2\_6

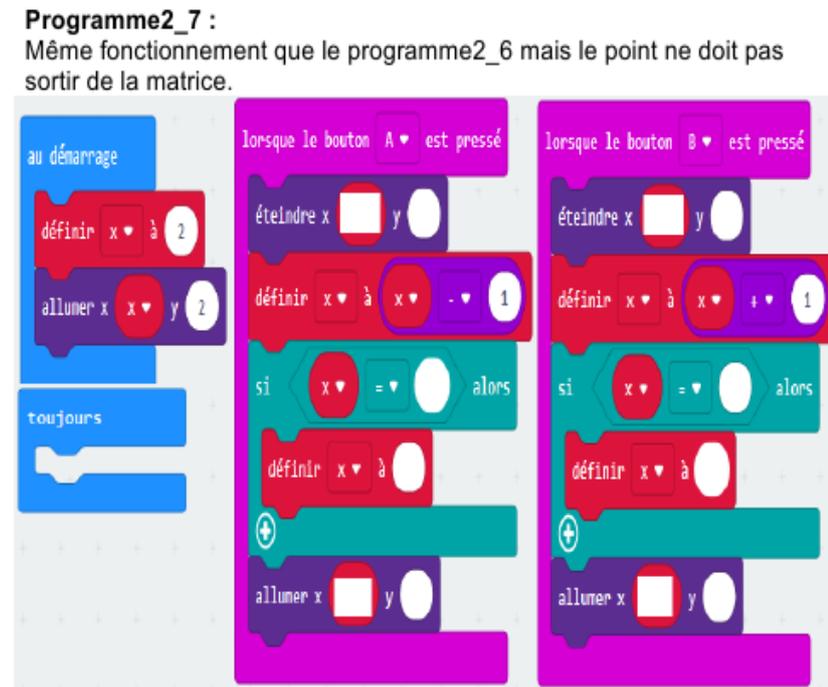
Le programme2\_7

Compléter les programmes  
Tester les programmes



**Programme2\_6 :**  
Au démarrage, positionner un point au centre  
Appui sur A : le point se déplace à gauche  
Appui sur B : le point se déplace à droite

The code for Programme2\_6 consists of three event-driven blocks. The 'au démarrage' block contains 'définir x à 0' and 'allumer x y'. The 'lorsque le bouton A est pressé' block contains 'éteindre x y', 'définir x à x - 1', and 'allumer x y'. The 'lorsque le bouton B est pressé' block contains 'éteindre x y', 'définir x à x + 1', and 'allumer x y'.



**Programme2\_7 :**  
Même fonctionnement que le programme2\_6 mais le point ne doit pas sortir de la matrice.

The code for Programme2\_7 is similar to Programme2\_6 but includes boundary checks. The 'au démarrage' block sets 'x' to 2 and 'allumer x y' to 2. The 'lorsque le bouton A est pressé' block has a 'si x = 0 alors' condition that sets 'x' to 0 before decreasing it. The 'lorsque le bouton B est pressé' block has a 'si x = 2 alors' condition that sets 'x' to 2 before increasing it.

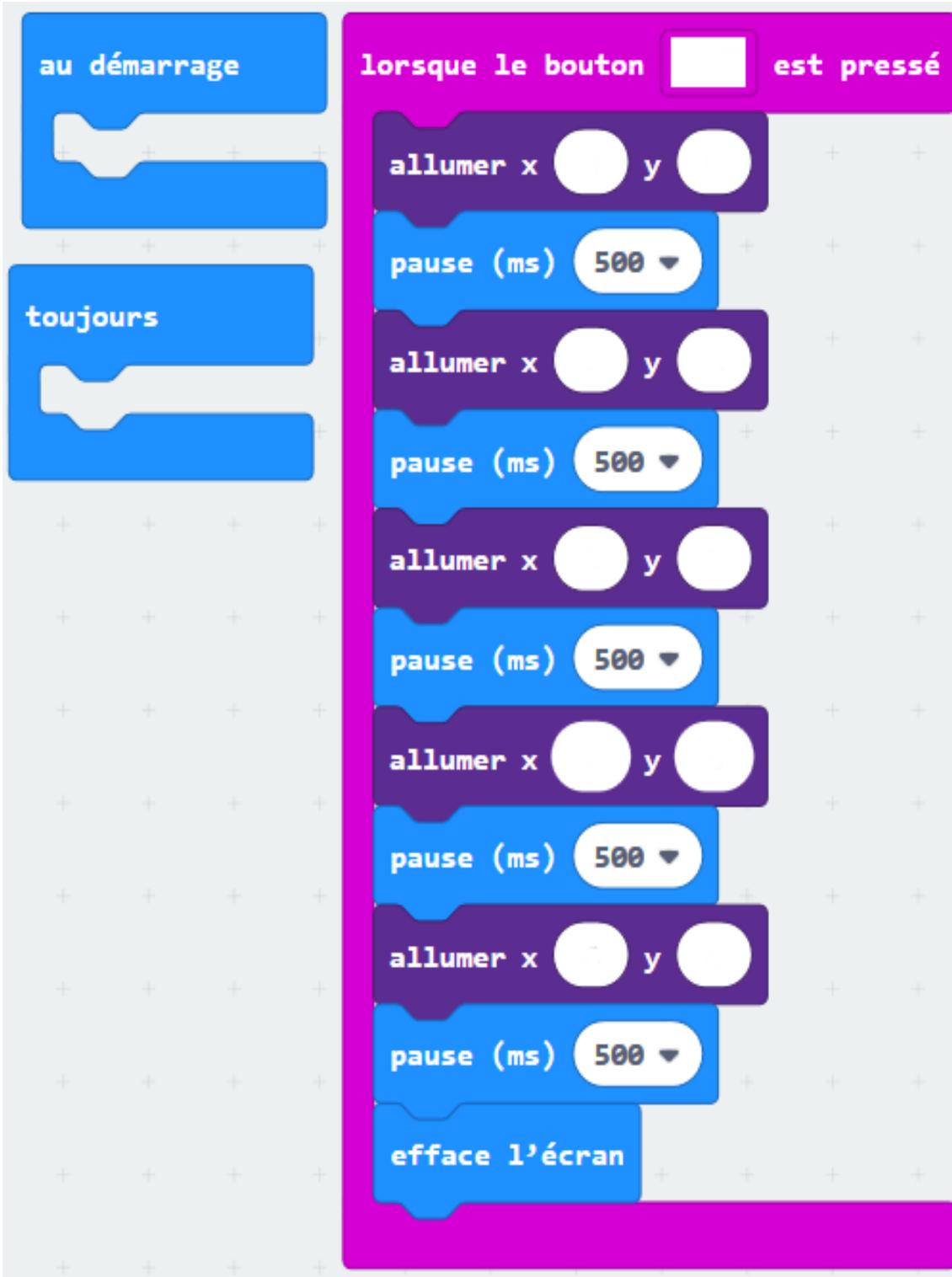
# boite à outil LED

Rechercher... 🔍

- Base
- Entrées
- Musique
- LED**
- plus
- Radio
- Boucles
- Logique

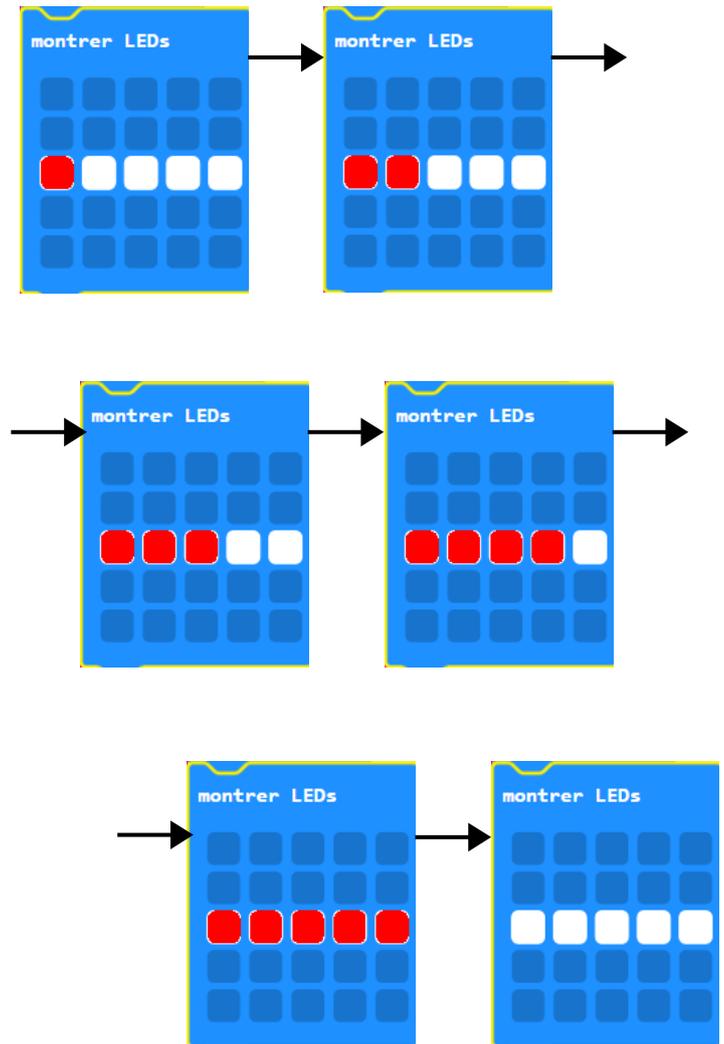
**LED**

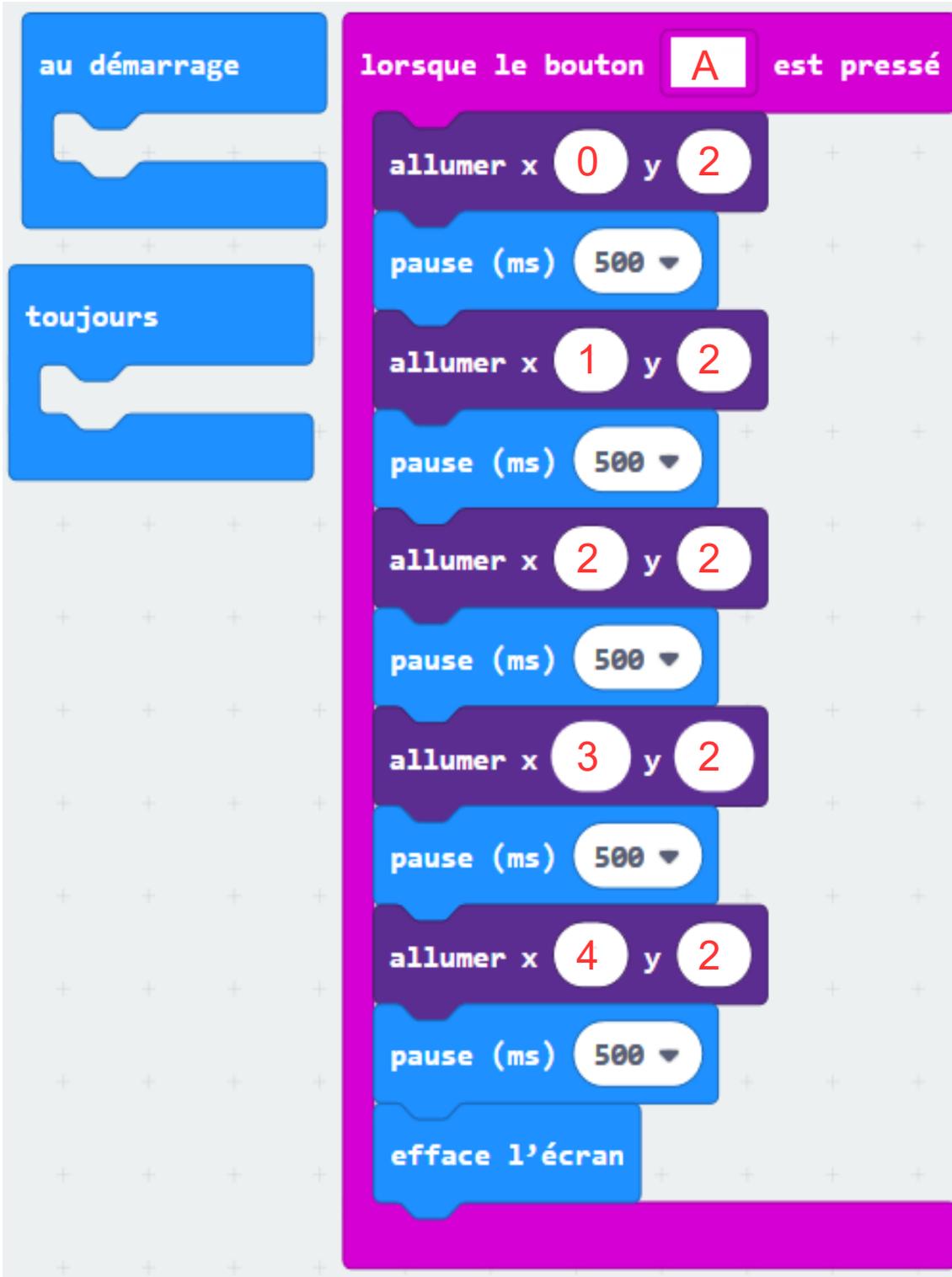
- allumer x 0 y 0 → Allume la LED en x,y
- éteindre x 0 y 0 → Eteint la LED en x,y
- activer/désactiver x 0 y 0 → Allume la LED en x,y si elle est éteinte  
Eteint la LED en x,y si elle est allumée
- allumée à x 0 y 0 → Permet de tester si la LED est allumée
- tracer le graphe de 0 à 0 → Permet de réaliser une fonction « vumètre »



## Programme2\_1 :

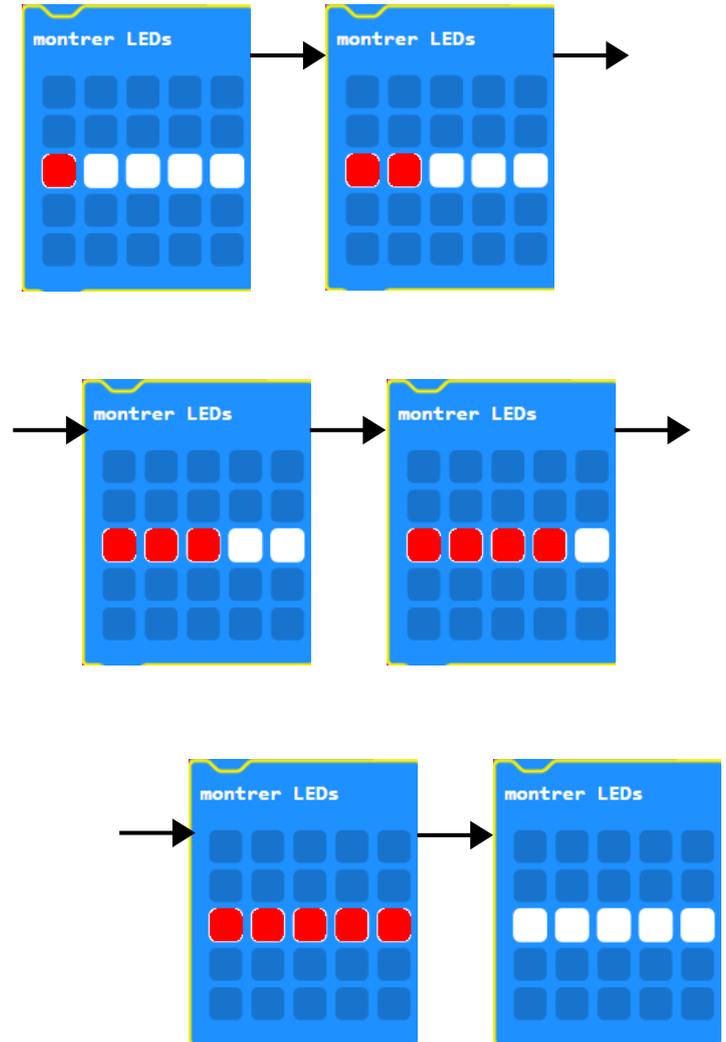
Quant A est pressé  
Permet de tracer une ligne au  
centre de la matrice puis effacer  
l'écran.





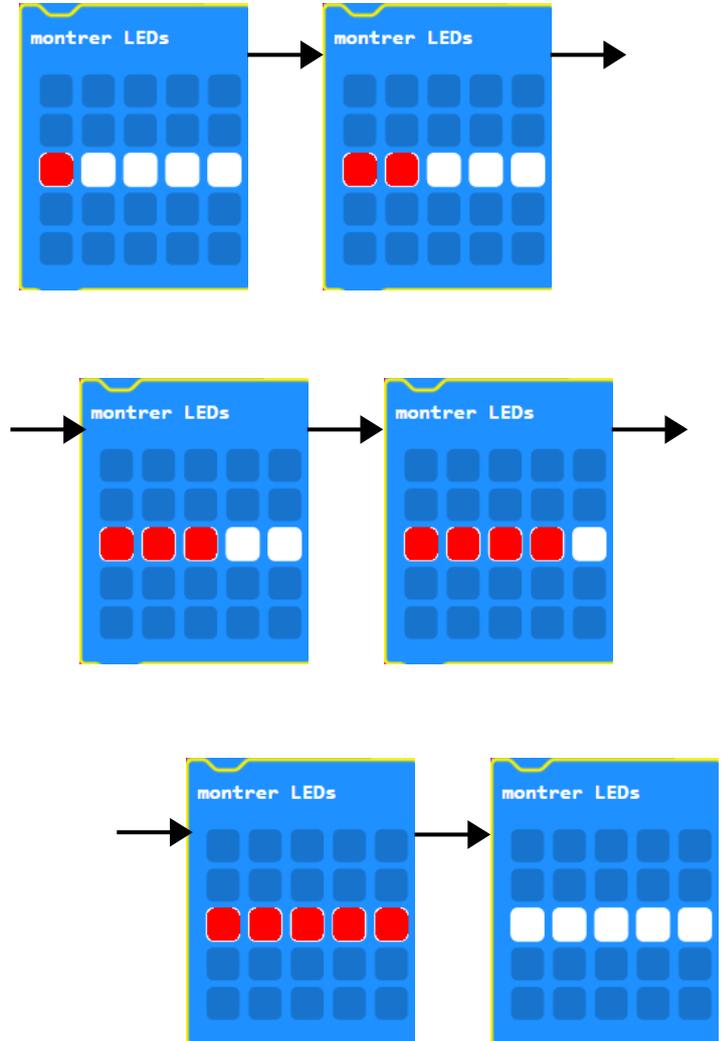
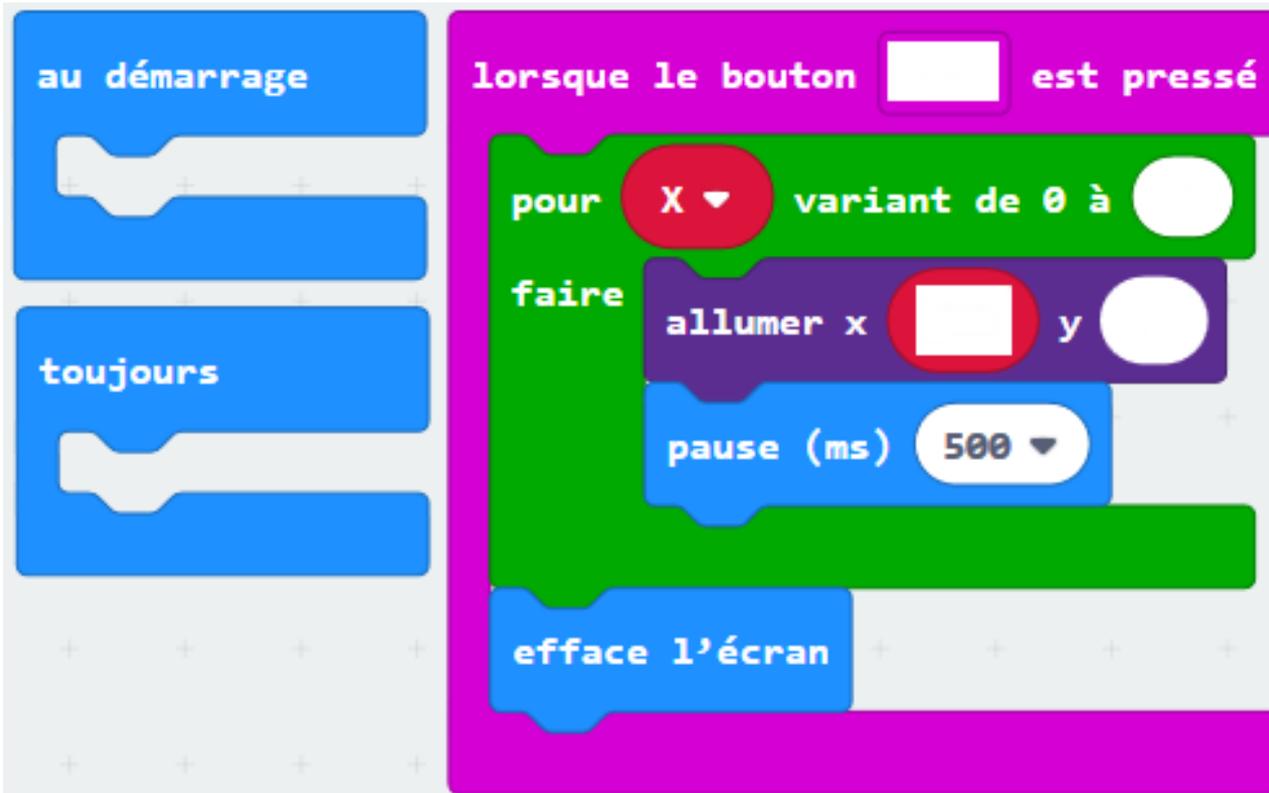
## Programme2\_1 :

Quant A est pressé  
Permet de tracer une ligne au centre de la matrice puis effacer l'écran.



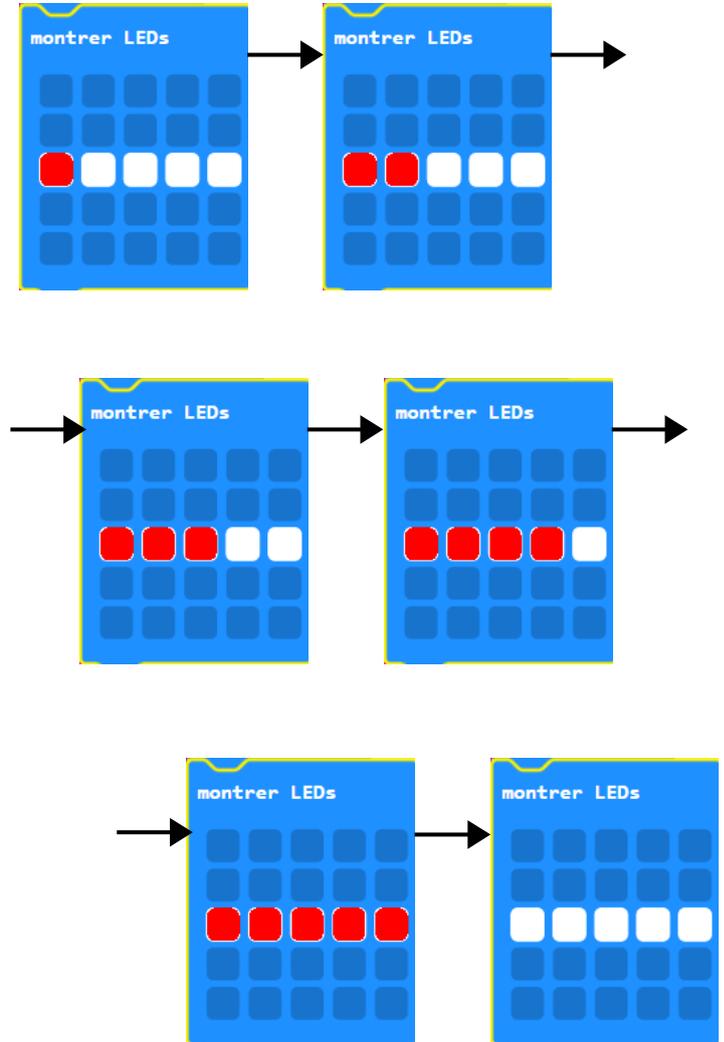
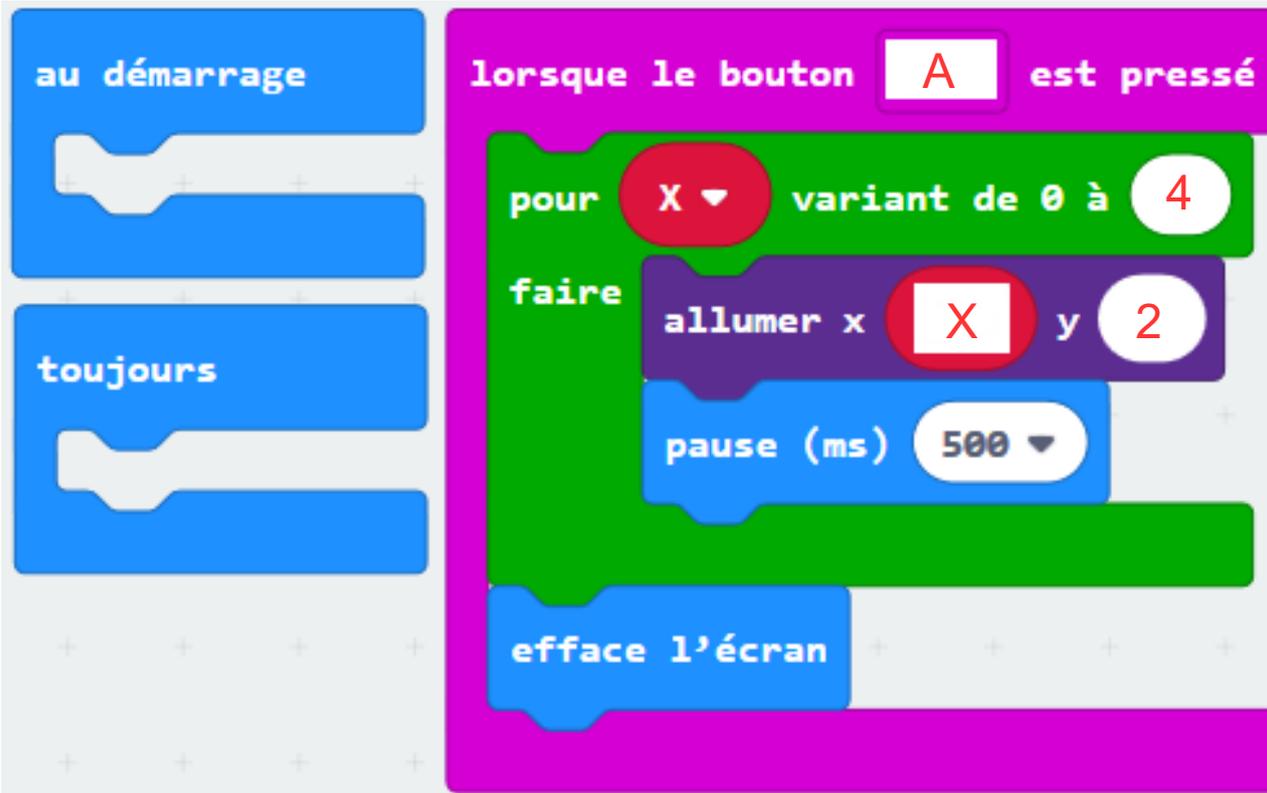
## Programme2\_2 :

Même fonctionnement que Programme2\_1 mais utilisant la variable X

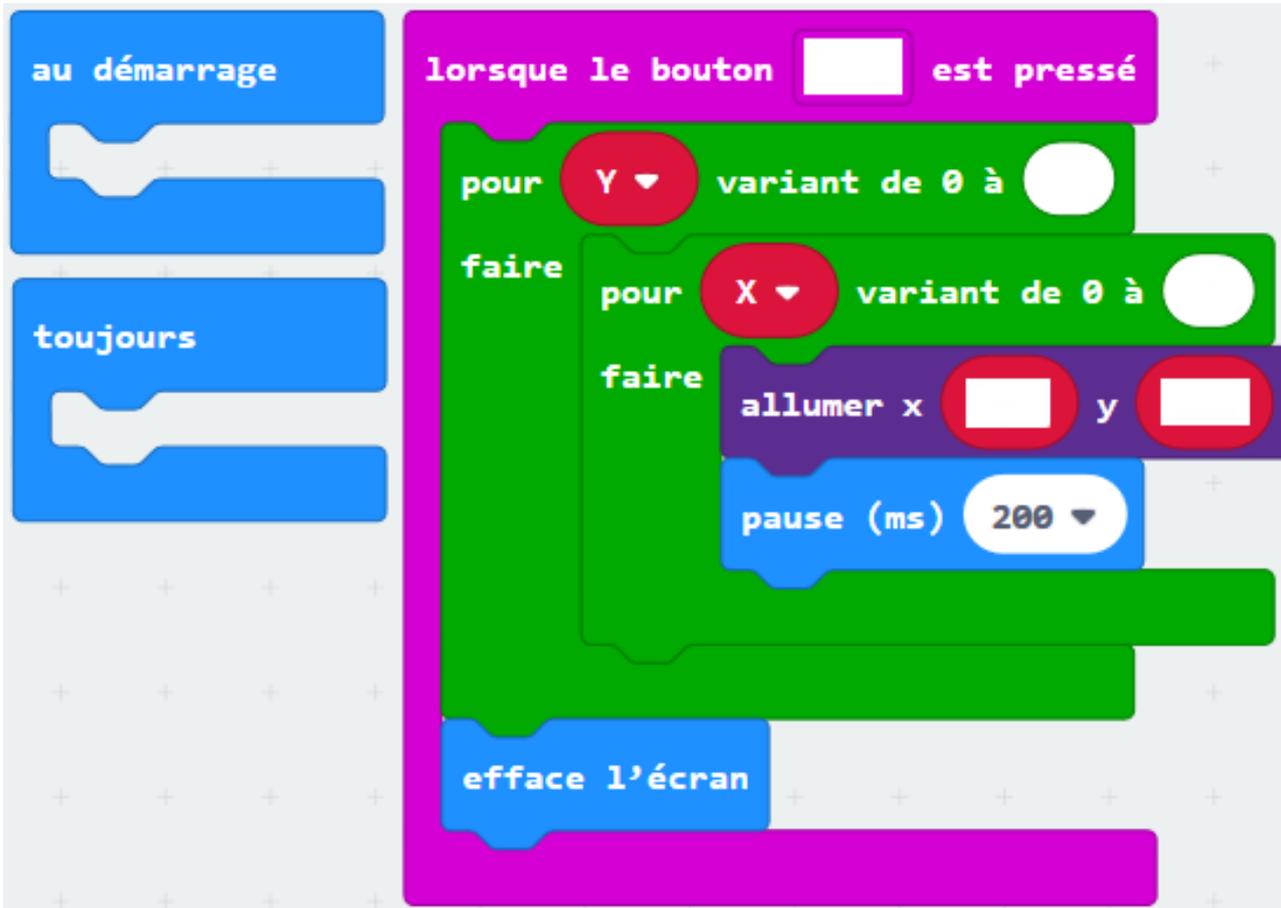
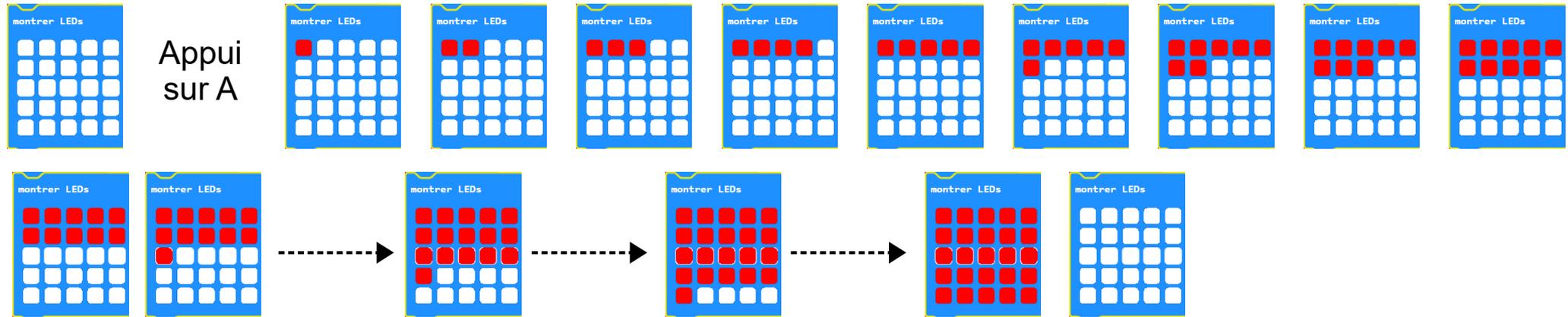


## Programme2\_2 :

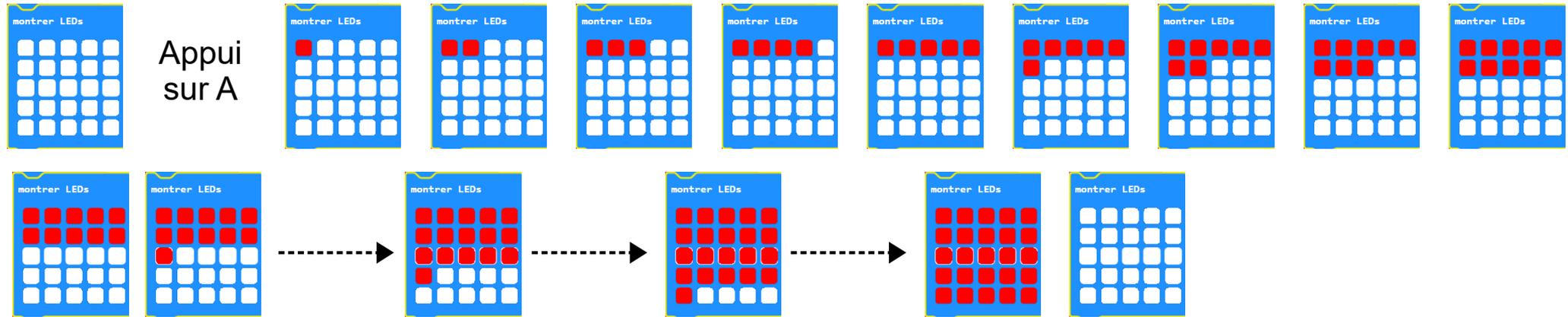
Même fonctionnement que Programme2\_1 mais utilisant la variable X



# Programme2\_3 :

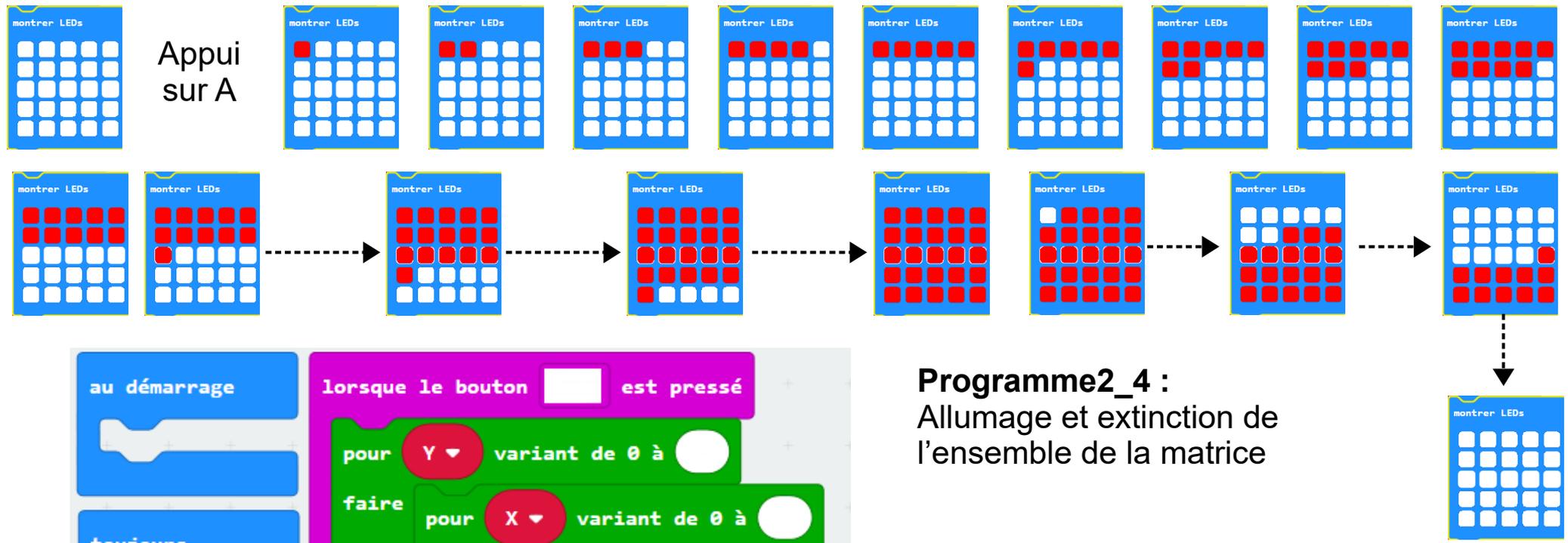


# Programme2\_3 :



```
au démarrage
toujours

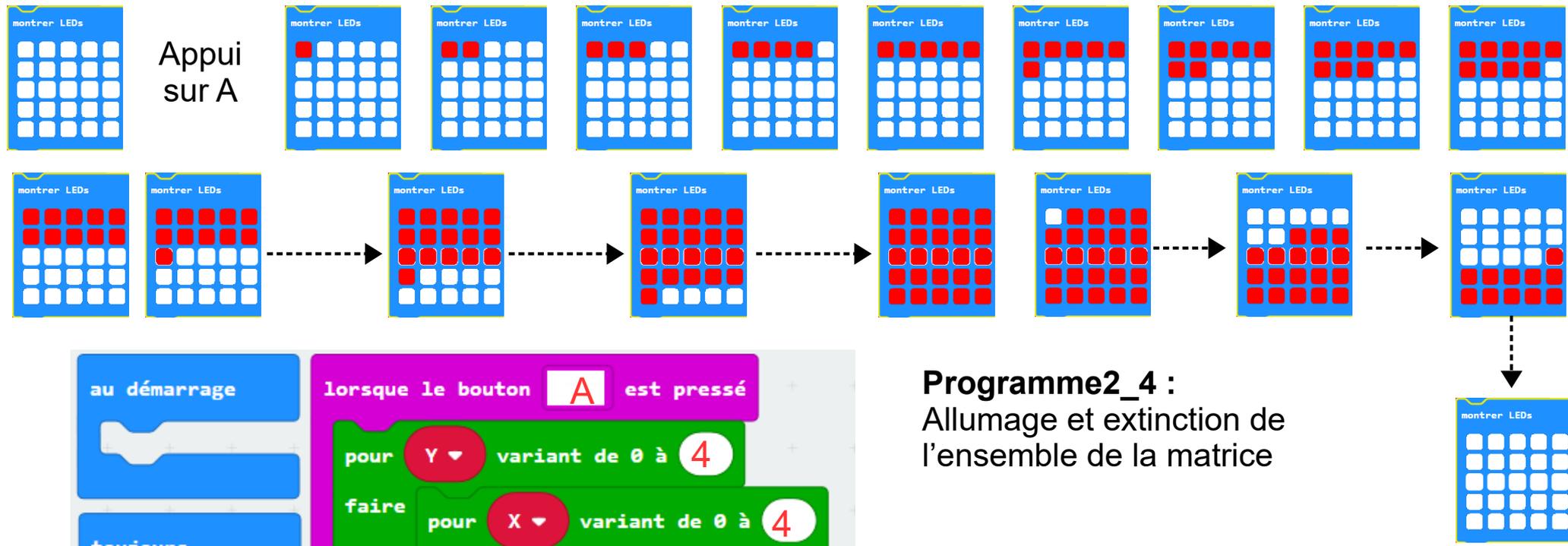
lorsque le bouton A est pressé
  pour Y variant de 0 à 4
    faire
      pour X variant de 0 à 4
        faire
          allumer x X y Y
          pause (ms) 200
      efface l'écran
```



**Programme2\_4 :**  
Allumage et extinction de l'ensemble de la matrice

```

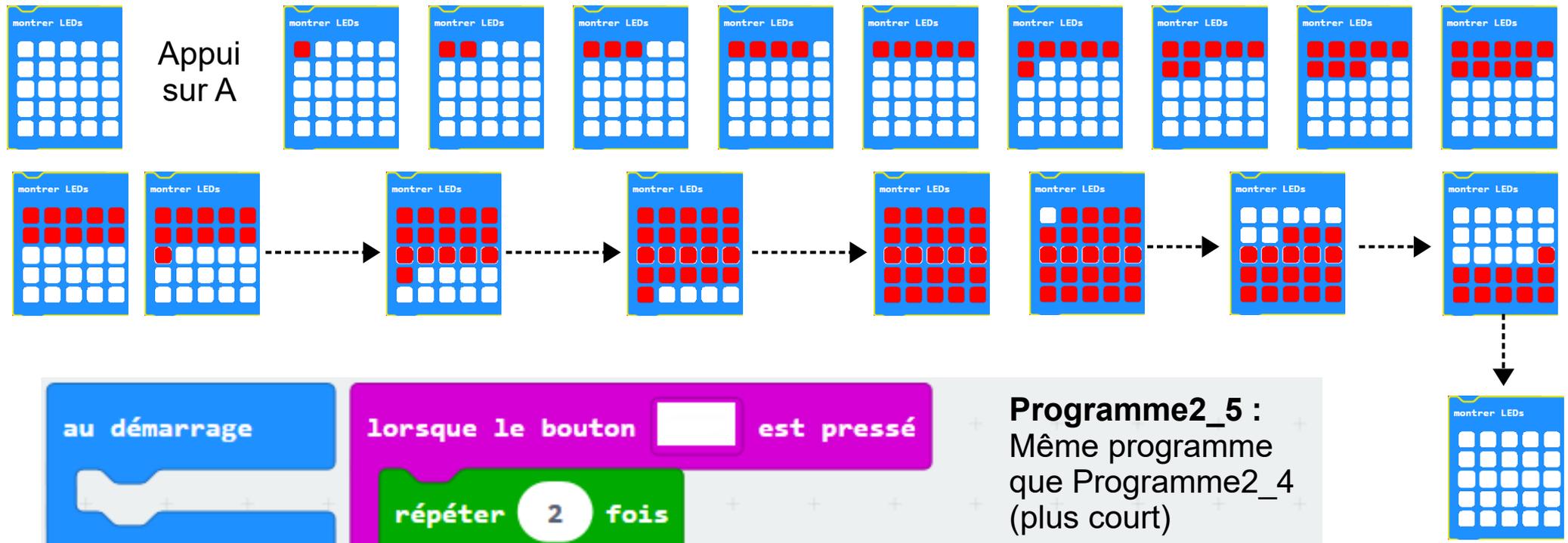
au démarrage
  toujours
    lorsque le bouton [ ] est pressé
      pour Y variant de 0 à 4
        faire
          pour X variant de 0 à 4
            faire
              allumer x [ ] y [ ]
              pause (ms) 200
      pour Y variant de 0 à 4
        faire
          pour X variant de 0 à 4
            faire
              éteindre x [ ] y [ ]
              pause (ms) 200
  
```



**Programme2\_4 :**  
Allumage et extinction de l'ensemble de la matrice

```

au démarrage
  toujours
  lorsque le bouton A est pressé
    pour Y variant de 0 à 4
      faire
        pour X variant de 0 à 4
          faire
            allumer x X y Y
            pause (ms) 200
    pour Y variant de 0 à 4
      faire
        pour X variant de 0 à 4
          faire
            éteindre x X y Y
            pause (ms) 200
  
```



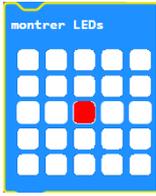
```

au démarrage
  toujours
    lorsque le bouton [ ] est pressé
      répéter 2 fois
        faire
          pour Y variant de 0 à [ ]
            faire
              pour X variant de 0 à [ ]
                faire
                  activer/désactiver x [ ] y [ ]
                  pause (ms) 200
  
```

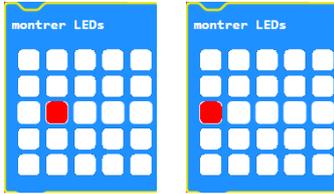
**Programme2\_5 :**  
Même programme  
que Programme2\_4  
(plus court)



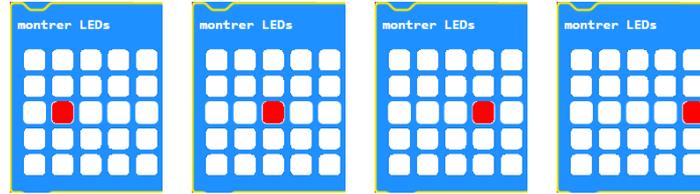
Démarrage



Appui sur A ←



Appui sur B →



Déclarer la variable x

au démarrage

définir x à [ ]

allumer x [ ] y [ ]

toujours

lorsque le bouton A est pressé

éteindre x [ ] y [ ]

définir x à [ ] - 1

allumer x [ ] y [ ]

lorsque le bouton B est pressé

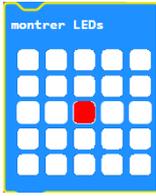
éteindre x [ ] y [ ]

définir x à [ ] + 1

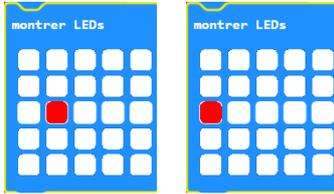
allumer x [ ] y [ ]

**Programme2\_6 :**  
 Au démarrage, positionner un point au centre  
 Appui sur A : le point se déplace à gauche  
 Appui sur B : le point se déplace à droite

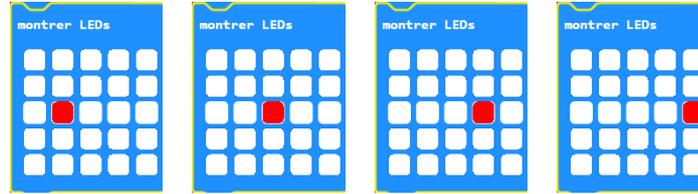
Démarrage



Appui sur A ←



Appui sur B →



Déclarer la variable x

```

au démarrage
  définir x à 2
  allumer x x y 2

toujours

```

**Programme2\_6 :**  
 Au démarrage, positionner un point au centre  
 Appui sur A : le point se déplace à gauche  
 Appui sur B : le point se déplace à droite

```

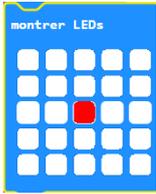
lorsque le bouton A est pressé
  éteindre x x y 2
  définir x à x - 1
  allumer x x y 2

lorsque le bouton B est pressé
  éteindre x x y 2
  définir x à x + 1
  allumer x x y 2

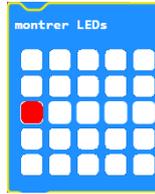
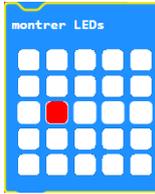
```

**Problème :**  
 Le point disparaît sur le côté de la matrice

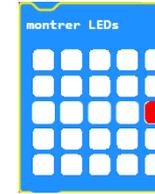
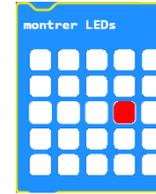
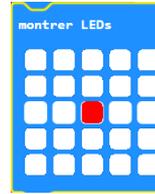
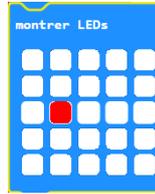
Démarrage



Appui sur A ←



Appui sur B →



### Programme2\_7 :

Même fonctionnement que le programme2\_6 mais le point ne doit pas sortir de la matrice

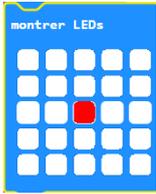
```

au démarrage
  définir x à 2
  allumer x x y 2
  toujours

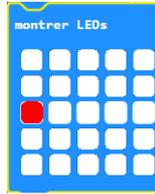
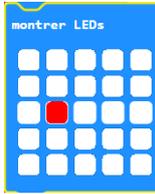
lorsque le bouton A est pressé
  éteindre x y
  définir x à x - 1
  si x = 0 alors
    définir x à 0
  allumer x y

lorsque le bouton B est pressé
  éteindre x y
  définir x à x + 1
  si x = 0 alors
    définir x à 0
  allumer x y
  
```

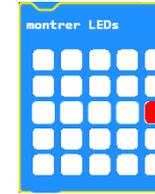
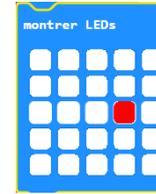
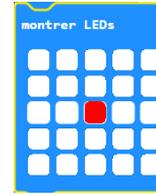
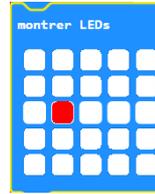
Démarrage



Appui sur A



Appui sur B



### Programme2\_7 :

Même fonctionnement que le programme2\_6 mais le point ne doit pas sortir de la matrice

```
au démarrage
  définir x à 2
  allumer x x y 2

toujours

lorsque le bouton A est pressé
  éteindre x X y 2
  définir x à x - 1
  si x = -1 alors
    définir x à 0
  +
  allumer x X y 2

lorsque le bouton B est pressé
  éteindre x X y 2
  définir x à x + 1
  si x = 5 alors
    définir x à 4
  +
  allumer x X y 2
```