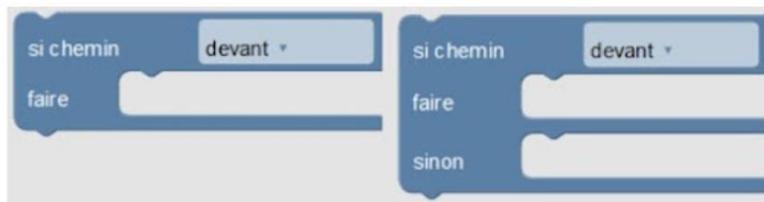


## Quiz 1° partie sur la notion d'algorithme

Les blocs ci-dessous sont

1 point



- des blocs de répétition
- des blocs de branchement conditionnel
- des instructions simples

Un algorithme

1 point

- est une suite finie d'instructions permettant de résoudre un problème
- peut comprendre une liste infinie d'instructions
- peut être imprécis
- peut être ambigu

Cocher les 3 instructions ou structures qui permettent de contrôler l'enchaînement des opérations (avancer, tourner, ...) dans un algorithme.

1 point

- la séquence
- le branchement conditionnel (ou la condition)
- la variable
- la répétition ou l'itération

(Terminez la phrase suivante) La notion de séquence dans un algorithme veut dire que les instructions sont exécutées \_\_\_\_\_.

1 point

- dans un ordre quelconque choisi par le programme
- toutes en même temps
- dans l'ordre où elles sont écrites dans la séquence

Comment s'appelle l'instruction ou la structure de contrôle qui permet d'exécuter

1 point

1. une séquence d'instructions si une condition est satisfaite,
2. une autre séquence dans le cas contraire ?

- la séquence
- la procédure
- la condition, (ou le branchement conditionnel)
- la répétition ou itération

En général dans un programme, à quoi sert une structure ou bloc de type 'si/ sinon' ?

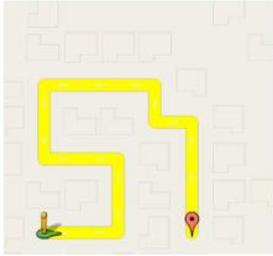
1 point

(choisir les 2 réponses qui correspondent le mieux)

- à arrêter l'application
- à répéter une séquence d'instructions.
- à vérifier si une condition est vraie ou fausse.
- à dire à l'application quel chemin à suivre dans le code



Cochez tous les algorithmes qui permettent au personnage d'arriver à l'icône rouge 1 point



```

répéter jusqu'à
faire
  si chemin vers la droite ☐
  faire tourner à droite ☐
si chemin vers la droite ☐
faire tourner à gauche ☐
avancer
  
```

Option 1

```

répéter jusqu'à
faire
  si chemin vers la gauche ☐
  faire tourner à gauche ☐
  si chemin vers la droite ☐
  faire tourner à droite ☐
  avancer
  
```

Option 2

```

répéter jusqu'à
faire
  si chemin vers la droite ☐
  faire tourner à droite ☐
  si chemin vers la gauche ☐
  faire tourner à gauche ☐
  avancer
  
```

Option 3

Cochez tous les algorithmes qui permettent au personnage d'atteindre la sortie (point rouge) 0 point



```

répéter jusqu'à
faire
  si chemin devant ☐
  faire avancer ☐
  si chemin vers la gauche ☐
  faire tourner à gauche ☐
  
```

Algo 1

```

répéter jusqu'à
faire
  si chemin devant ☐
  faire avancer ☐
  si chemin vers la gauche ☐
  faire tourner à gauche ☐
  si chemin vers la droite ☐
  faire tourner à droite ☐
  
```

Algo 2

```

répéter jusqu'à
faire
  si chemin devant ☐
  faire avancer ☐
  sinon tourner à gauche ☐
  avancer
  
```

Algo 3

```

répéter jusqu'à
faire
  si chemin devant ☐
  faire avancer ☐
  sinon tourner à gauche ☐
  
```

Algo 4

Cochez tous les circuits où cet algorithme permet au bonhomme d'atteindre la sortie (point rouge) 0 point

```

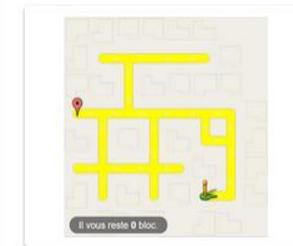
répéter jusqu'à
faire
  si chemin devant ☐
  faire avancer ☐
  sinon
    si chemin vers la gauche ☐
    faire tourner à gauche ☐
    sinon tourner à droite ☐
  
```



N4



N8



N9



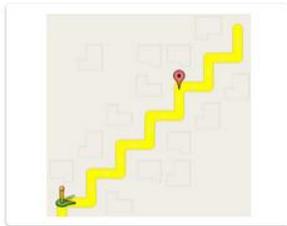
N10

### Quiz 3° partie – bonus pour les rapides

Cochez tous les circuits où cet algorithme permet au bonhomme d'atteindre la sortie (point rouge) 0 point

```

répéter jusqu'à
faire
  si chemin vers la gauche
  faire
    tourner à gauche
  sinon
    si chemin devant
    faire
      si chemin vers la droite
      sinon
        faire
          tourner à droite
        sinon
          tourner à droite
    avancer
  
```



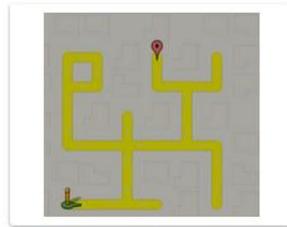
N4



N8



N9



N10

Le programme ci-dessous déplace un robot dans une grille. Le robot est représenté par un triangle orienté vers la gauche. Il peut avancer dans les cases blanches ou grises, mais pas dans les noires. L'objectif (butAtteint) est que le robot arrive jusqu'à la case grise. Cochez le cas où le robot n'y arrive PAS. 1 point

```

REPETER JUSQU'A (ButAtteint())
{
  SI (peut_aller(enAvant))
  {
    Avancer()
  }
  SI (peut_aller(aDroite))
  {
    tournerADroite()
  }
  SI (peut_aller(aGauche))
  {
    tournerAGauche()
  }
}
  
```



①



②



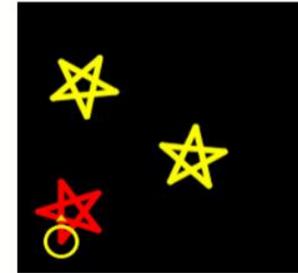
③



④

- Option 1
- Option 2
- Option 3
- Option 4

Au départ, le crayon est posé sur le point le plus bas de l'étoile rouge et orienté vers le haut. Le côté des étoiles mesure 50. Quel est l'algorithme qui permet d'obtenir ce dessin ? 1 point



```

mettre la couleur à
répéter 3 fois
faire
  mettre la couleur à
  répéter 5 fois
  faire
    avancer de 50
    tourner à droite de 144
  lever le crayon
  avancer de 150
  tourner à droite de 120
  poser le crayon
  
```

algo 1

```

mettre la couleur à
répéter 3 fois
faire
  répéter 5 fois
  faire
    avancer de 50
    tourner à droite de 90
  mettre la couleur à
  lever le crayon
  avancer de 150
  tourner à droite de 120
  poser le crayon
  
```

algo 2

```

mettre la couleur à
répéter 3 fois
faire
  répéter 5 fois
  faire
    avancer de 50
    tourner à droite de 144
  lever le crayon
  avancer de 150
  mettre la couleur à
  tourner à droite de 120
  poser le crayon
  
```

algo 3

```

mettre la couleur à
répéter 3 fois
faire
  répéter 3 fois
  faire
    avancer de 50
    tourner à droite de 144
  mettre la couleur à
  lever le crayon
  avancer de 150
  tourner à droite de 120
  poser le crayon
  
```

algo 4