



# EXERCICES

## Organigramme et logigramme

Niveau 1 à 3

Nom :

Prénom :

**CT1.3 / CT2.5 / CT2.7 :** Imaginer des solutions pour produire des objets et des éléments de programmes informatiques en réponse au besoin.

**CT3.1 :** Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux.

**CT4.2 / CT5.5 :** Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Exo 1



Réalise la description par algorithme permettant à Mario d'avancer dès que le joueur appuie sur le bouton de la manette de jeu.



EVÈNEMENTS

ACTIONS

Bouton Avancer activé ?

Avancer de 10 pixels

Exo 2



Réalise la description par algorithme permettant de gérer le stock de HandSpinner mis en rayon.

Quand la quantité est inférieure à 50, une commande se fait automatiquement auprès du fournisseur.

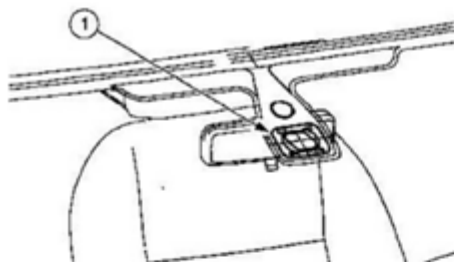
EVÈNEMENTS

ACTIONS

Quantité < 50 ?

Passer une commande

Exo 3



Réalise la description par algorithme des essuie-glaces de voiture automatique.

Un capteur permet de déterminer s'il pleut. Dès la détection de la pluie, les essuie-glaces s'activent.

EVÈNEMENTS

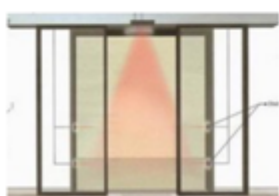
ACTIONS

Détection pluie ?

Activer essuie-glaces

Désactiver essuie-glaces

Exo 4



Réalise la description par algorithme permettant, dès la détection d'une personne, l'ouverture de la porte pendant 5 secondes.

EVÈNEMENTS

ACTIONS

Détection présence ?

Ouvrir porte

Fermer porte

Attendre 5 secondes

## NIVEAU 2

### Exo 1



Réalise l'algorithme permettant le fonctionnement connu d'un radar de vitesse automatique : Un flash apparaît dès la détection d'un véhicule à une vitesse plus grande que 90 km/h afin de prendre la photo de la plaque pour l'envoyer au central.

#### EVÉNEMENTS

#### ACTIONS

Détection véhicule avec une vitesse > 90km/h ?	Lecture plaque
	Activer Flash
	Envoyer information au central

### Exo 2



Réalise l'algorithme permettant, dès la détection d'une personne la mise en marche d'un escalator de bas en haut. Afin de limiter la consommation d'énergie, l'escalator est initialement (au début) à l'arrêt et ne fonctionne pas tant qu'une personne n'est pas détectée. Il faut 1 minute à l'escalator pour monter une personne.

#### EVÉNEMENTS

#### ACTIONS

Détection bas ?	Activer Escalator
	Désactiver Escalator
	Attendre 1 minute

### Exo 3

Suite de l'exercice précédent.

En montant les marches, la personne qui utilise l'escalator mettra moins de temps pour arriver en haut. Inutile donc dans ce cas de faire fonctionner l'escalator durant toute une minute.

Pour des soucis d'économie d'énergie propose un autre algorithme permettant le même principe, mais cette fois-ci, l'escalator devra s'arrêter uniquement lorsque la personne qui monte est détectée en haut de l'escalator.

Détection Haut



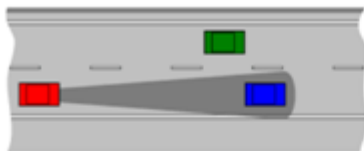
Détection Bas

#### EVÉNEMENTS

#### ACTIONS

Détection bas ?	Activer Escalator
Détection haut ?	Désactiver Escalator

### Exo 4



Imaginons un système embarqué dans une voiture capable d'évaluer si la distance entre la voiture et celle de devant est dangereuse. Si la distance devient dangereuse un voyant vert passe au rouge.

La sécurité routière impose une distance de sécurité de 2 bandes blanches.

#### EVÉNEMENTS

#### ACTIONS

Variable_distance < 2 bandes blanches ?	Enregistrer mesure dans « Variable_distance »
	Mesure distance
	Allumer voyant Vert
	Allumer voyant Rouge

## NIVEAU 3

### Exo 1



Propose un algorithme définissant le fonctionnement simple d'une cafetière :

Le café ne peut couler que s'il y a de l'eau présente dans le réservoir et que si l'eau est à suffisamment chauffée par la cafetière (température  $> 88^{\circ}$ ).

#### EVÈNEMENTS

#### ACTIONS

Bouton café activé ?	Chauffer eau
T° eau $> 88^{\circ}$ ?	Indiquer « ajouter de l'eau »
Eau présente ?	Servir café

### Exo 2



Un hôtel est équipé d'un système semi-automatique de contrôle d'accès.

De 8h00 à 22h00 le concierge en apercevant un client à l'entrée actionne un bouton pour permettre l'accès au client. De 22h00 à 8h00, l'hôtel n'ayant pas de veilleur de nuit, les clients déverrouillent la porte d'entrée de l'hôtel via un digicode à l'extérieur.

Propose une description du fonctionnement (algorithme ou algorithme).

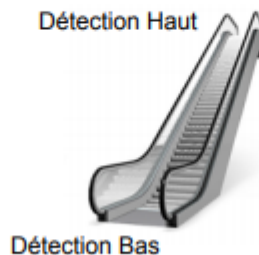
#### EVÈNEMENTS

#### ACTIONS

Bouton concierge activé ?	Déverrouiller porte
8h00 < présence < 22h00	Verrouiller porte
22h00 < Présence < 8h00	Attendre 5 secondes
Digicode correct ?	

### Exo 3

Afin de limiter la consommation énergétique, l'Escalator est à l'arrêt lorsque personne ne l'utilise. Dès la détection d'une personne en bas, l'Escalator se met en fonctionnement. Propose une description du fonctionnement.



#### EVÈNEMENTS

#### ACTIONS

Détection bas ?	Escalator OFF
Détection haut ?	Escalator ON
Compteur = 0 ?	Compteur = Compteur + 1
	Compteur = Compteur - 1
	Compteur = 0

**Exo 4** Un parking utilise une caméra avec un logiciel de lecture de plaques d'immatriculation afin de limiter les problèmes de perte de ticket.

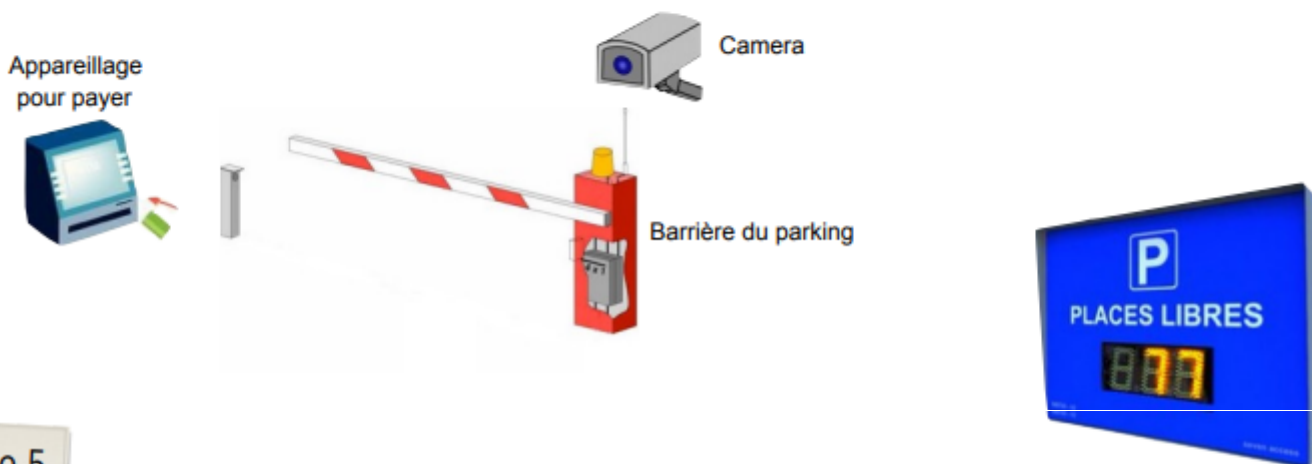
Lorsque l'utilisateur entre dans le parking, la lecture/enregistrement de la plaque s'effectue (variable\_plaque) ainsi que l'enregistrement de l'heure d'entrée (variable\_heure\_entree).

Lorsque l'utilisateur sort du parking, il paye directement à la barrière au moment où il est reconnu par le système de la camera.

#### EVÈNEMENTS

#### ACTIONS

Détection entrée parking ?	Lecture/Enregistrement plaque dans « variable_plaque »
Détection sortie parking ?	Enregistrement de l'heure dans « variable_h_entree »
Voiture passée ?	Lecture plaque
Parking payé ?	Variable_Prix = (heure actuelle - variable_heure_entree) x 1€
	Afficher Variable_prix
	Ouverture barrière
	Fermeture barrière



**Exo 5** Amélioration du fonctionnement.

Afin de limiter le problème de recherche de place, le parking est équipé d'un afficheur à l'extérieur permettant d'indiquer le nombre de places disponibles.

Le nombre de places disponibles est également envoyé sur un serveur, ce qui permet aux utilisateurs de l'application mobile d'être informé à distance.

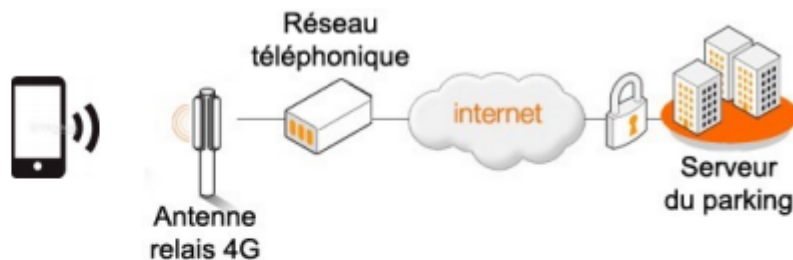
Améliorer le fonctionnement précédent.

Il y a 150 places dans le parking.

Utilisation de *Variable\_Place* pour déterminer le nombre de places disponibles.

#### ACTIONS

Envoyer Variable_Place sur serveur
Afficher Variable_Place sur afficheur



**Exo 6**

Propose la description du programme de l'application mobile.

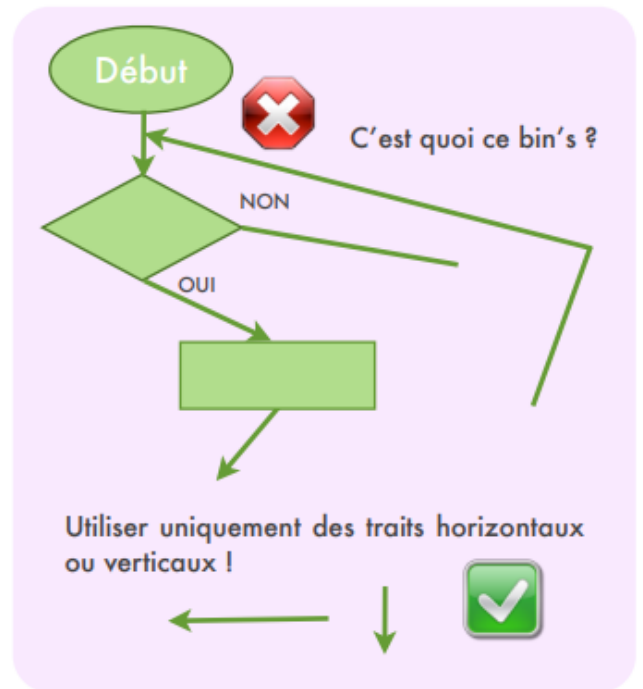
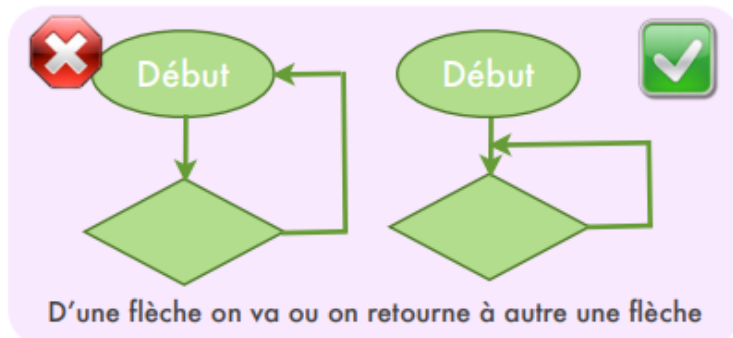
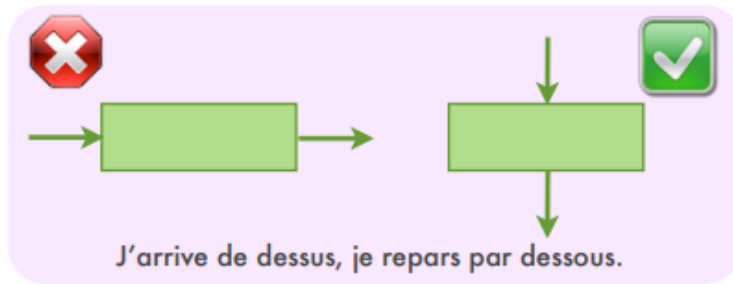
Dans le cas où il n'y a plus de place dans le parking, un message affiche en rouge « Complet ». Dans le cas contraire, l'écran est vert avec comme indication le nombre de places disponibles.

#### ACTIONS

Récupérer Variable_Place sur serveur
--------------------------------------

# A RETENIR !

## Attention aux erreurs !



## A vérifier à chaque fois !

