

# QCM – Les Automates programmables industriels

Nom : ..... Prénom : .....

Classe : ..... Date : .....

- **Consigne :**

Coche la bonne réponse lorsque la question ne comporte qu'une seule réponse correcte.

Coche toutes les bonnes réponses lorsque plusieurs réponses sont possibles (mention « plusieurs réponses » dans l'énoncé).

**Q1.** L'abréviation « API » signifie Automate Programmable Industriel.

- a) Vrai
- b) Faux

**Q2.** Dans un système automatisé moderne, l'API joue le rôle de :

- a) Capteur
- b) Actionneur
- c) Unité de traitement
- d) Préactionneur

**Q3.** Quels éléments font partie de l'architecture interne d'un API moderne ? (plusieurs réponses possibles)

- a) CPU
- b) Carte graphique
- c) Mémoire RAM / Flash
- d) Horloge interne
- e) Bus interne
- f) Module de communication Ethernet

**Q4.** La fonction principale d'un API est :

- a) Lire les entrées, exécuter un programme, mettre à jour les sorties
- b) Commander directement les moteurs de puissance
- c) Remplacer les capteurs
- d) Générer uniquement des signaux analogiques

**Q5.** Le microprocesseur d'un API moderne :

- a) Exécute les instructions du programme utilisateur
- b) Gère les communications réseau
- c) Réalise les fonctions logiques et séquentielles
- d) Toutes les réponses précédentes

**Q6.** Les échanges entre CPU, mémoire et modules E/S se font via :

- a) Un bus interne
- b) Une liaison USB
- c) Une liaison série RS232
- d) Une liaison radio

**Q7.** Les informations échangées entre CPU et modules E/S sont :

- a) Analogiques
- b) Numériques
- c) Pneumatiques
- d) Hydrauliques

**Q8.** La mémoire d'un API moderne permet de stocker (plusieurs réponses possibles) :

- a) Le programme utilisateur
- b) Les valeurs des entrées/sorties
- c) Les variables internes
- d) Les données de configuration

**Q9.** La RAM d'un API est une mémoire volatile.

- a) Vrai
- b) Faux

**Q10.** Une mémoire Flash peut être réécrite.

- a) Vrai
- b) Faux

**Q11.** Les notations modernes des entrées et sorties selon IEC 61131-3 sont :

- a) I et O
- b) %I et %Q
- c) E et S
- d) IN et OUT

**Q12.** Le langage Ladder (LD) permet :

- a) De représenter des équations logiques sous forme de contacts et bobines
- b) De programmer en texte structuré
- c) De créer des blocs fonctionnels
- d) De programmer des transitions de Grafcet

**Q13.** Le langage SFC (Grafcet) permet (plusieurs réponses possibles) :

- a) De représenter des séquences d'étapes et transitions
- b) De programmer des équations logiques
- c) De gérer des actions associées aux étapes
- d) De gérer des réceptivités

**Q14.** En notation IEC, une entrée numérique est identifiée par :

- a) %I
- b) %Q
- c) %M
- d) %T

**Q15.** En notation IEC, une sortie numérique est identifiée par :

- a) %I
- b) %Q
- c) %M
- d) %MW

**Q16.** Les temporisations (timers) sont identifiées par :

- a) %T
- b) %MW
- c) %C
- d) %I

**Q17.** Les compteurs sont identifiés par :

- a) %C
- b) %T
- c) %Q
- d) %M

**Q18.** Les mots mémoire (%MW) servent notamment à :

- a) Stocker des valeurs analogiques
- b) Stocker des entiers
- c) Stocker des paramètres
- d) Toutes les réponses précédentes

**Q19.** Le cycle d'un API moderne suit l'ordre :

- a) Exécution → Lecture → Mise à jour
- b) Lecture → Exécution → Mise à jour
- c) Mise à jour → Lecture → Exécution
- d) Lecture → Mise à jour → Exécution

**Q20.** Unity Pro / EcoStruxure Control Expert permet de programmer (plusieurs réponses possibles) :

- a) En Ladder (LD)
- b) En Texte structuré (ST)
- c) En Grafcet (SFC)
- d) En blocs fonctionnels (FBD)

**Q21.** Les API modernes peuvent communiquer via :

- a) Ethernet
- b) Modbus TCP
- c) CANopen
- d) Toutes les réponses précédentes

**Q22.** Une sortie d'API commande généralement :

- a) Directement un moteur
- b) Un préactionneur (relais, contacteur)
- c) Un capteur
- d) Une alimentation

**Q23.** Les entrées analogiques permettent de lire :

- a) Des valeurs continues (0–10 V, 4–20 mA)
- b) Des états logiques 0/1
- c) Des signaux pneumatiques
- d) Des signaux hydrauliques

**Q24.** Les sorties analogiques permettent :

- a) De générer un signal 0–10 V ou 4–20 mA
- b) De commander un contacteur
- c) De lire un capteur
- d) De mesurer une tension

**Q25.** En SFC, les actions sont associées :

- a) Aux étapes
- b) Aux transitions
- c) Aux entrées
- d) Aux sorties

**Q26.** En SFC, les transitions sont franchies lorsque :

- a) Les actions sont actives
- b) Les réceptivités sont vraies
- c) Les sorties sont activées
- d) Le cycle API redémarre

# Corrigé enseignant – QCM Automates programmables industriels

Q1 : **a** – API = Automate Programmable Industriel (PLC en anglais).

Q2 : **c** – L'API est l'unité de traitement / commande du système automatisé.

Q3 : **a, c, d, e, f** – CPU, RAM/Flash, horloge, bus et modules de communication ; pas de carte graphique.[beckassets.core.windows+1](#)

Q4 : **a** – Cycle typique : lecture des entrées → exécution du programme → mise à jour des sorties.[twcontrols+1](#)

Q5 : **d** – Le CPU exécute le programme, gère communications et logique séquentielle.[library.abb+1](#)

Q6 : **a** – Échanges internes via bus (backplane / bus interne).[nuovaelva+1](#)

Q7 : **b** – Les échanges CPU-E/S sont numériques (0/1 ou mots binaires).[nuovaelva](#)

Q8 : **a, b, c, d** – Programme, image E/S, variables, configuration sont stockés en mémoire.[divapps.parker+1](#)

Q9 : **a** – RAM = mémoire volatile (contenu perdu hors tension).

Q10 : **a** – La Flash est non volatile et réinscriptible.

Q11 : **b** – Adressage direct IEC : %I, %Q, %M, etc.[beckassets.core.windows+1](#)

Q12 : **a** – LD = diagramme à contacts (relais), équations logiques sous forme contacts/bobines.[wikipedia+1](#)

Q13 : **a, c, d** – SFC (Grafset IEC) : étapes, transitions, actions, réceptivités.[library.abb+1](#)

Q14 : **a** – %I = entrée numérique.[beckassets.core.windows+1](#)

Q15 : **b** – %Q = sortie numérique.[nuovaelva+1](#)

Q16 : **a** – %T souvent utilisé pour les temporisations (timers) selon les constructeurs.

Q17 : **a** – %C pour les compteurs.

Q18 : **d** – %MW = mot mémoire, pour entiers, analogiques, paramètres, etc.

Q19 : **b** – Cycle PLC : lecture (input sampling) → exécution → mise à jour des sorties.[jc-plc+1](#)

Q20 : **a, b, c, d** – Unity Pro / EcoStruxure Control Expert supporte LD, ST, FBD, SFC (et IL).[se+1](#)

Q21 : **d** – Les API modernes supportent Ethernet, Modbus TCP, CANopen, etc.

Q22 : **b** – Une sortie d'API commande en général un préactionneur (relais, contacteur), pas directement le moteur.

Q23 : **a** – Entrées analogiques : mesures 0–10 V, 4–20 mA, etc.

Q24 : **a** – Sorties analogiques : consignes 0–10 V, 4–20 mA vers variateurs, vannes, etc.

Q25 : **a** – En SFC, les actions sont associées aux étapes.

Q26 : **b** – Une transition est franchie lorsque sa réceptivité est vraie et les étapes amont actives.

- 
1. <https://nuovaelva.it/docs/Bosch%20Rexroth/Tecnologie%20e%20prodotti%20per%20il%20controllo%20del%20movimento/07230504.pdf>
  2. [https://en.wikipedia.org/wiki/IEC\\_61131-3](https://en.wikipedia.org/wiki/IEC_61131-3)
  3. [https://beckassets.blob.core.windows.net/product/readingsample/684471/9783642120145\\_excerpt\\_001.pdf](https://beckassets.blob.core.windows.net/product/readingsample/684471/9783642120145_excerpt_001.pdf)
  4. <https://www.se.com/ca/en/faqs/FA378550/>
  5. <https://twcontrols.com/rslogix-500-course/plc-scan-cycle>
  6. [https://en.jc-plc.com/News\\_xq/7942422.html](https://en.jc-plc.com/News_xq/7942422.html)
  7. <https://library.e.abb.com/public/81478a314e1386d1c1257b1a005b0fc0/2101127.pdf>
  8. [https://divapps.parker.com/divapps/emn/pdf/ACR/IEC61131-3\\_User\\_Guide.pdf](https://divapps.parker.com/divapps/emn/pdf/ACR/IEC61131-3_User_Guide.pdf)
  9. [https://pimassetsprd.blob.core.windows.net/assets/apc\\_Original/37/18/41993718.pdf](https://pimassetsprd.blob.core.windows.net/assets/apc_Original/37/18/41993718.pdf)
  10. [https://d1.amobbs.com/bbs\\_upload782111/files\\_31/ourdev\\_569653.pdf](https://d1.amobbs.com/bbs_upload782111/files_31/ourdev_569653.pdf)