

# SAS – Langage de Programmation C

## Day 1 : Exercices sur les Conditions

YouCode – 2025

---



## Table des Matières

- Challenge 1 : Test de Parité
- Challenge 2 : Identification de Voyelles
- Challenge 3 : Somme Conditionnelle
- Challenge 4 : Résolution d'Équation du Second Degré
- Challenge 5 : Convertisseur d'Années (Menu Interactif)
- Challenge 6 : Classification de Nombres
- Challenge 7 : Vérification d'Alphabet Majuscule
- Challenge 8 : Système de Mentions Académiques
- Challenge 9 : Analyse Complète de Caractères
- Challenge 10 : Reformatage de Dates
- Challenge 11 : Générateur de Jour Aléatoire
- Challenge 12 : Comparateur d'Heures
- Challenge 13 : Point sur un Segment (Géométrie)

## Challenge 1 : Test de Parité

Développez un programme qui détermine si un nombre entier saisi par l'utilisateur est pair ou impair.

**Règle :** Un nombre est pair si `nombre % 2 == 0`, sinon il est impair.

**Objectif :** Maîtriser l'opérateur modulo et les conditions simples.

---

## Challenge 2 : Identification de Voyelles

Créez un programme qui vérifie si un caractère saisi est une voyelle (a, e, i, o, u) en utilisant l'instruction `switch case`.

**Note :** Considérez les voyelles en minuscules et majuscules.

**Objectif :** Utiliser la structure `switch case` pour des comparaisons multiples.

---

## Challenge 3 : Somme Conditionnelle

Écrivez un programme qui calcule la somme de deux entiers avec une règle spéciale : si les deux nombres sont identiques, retournez le triple de leur somme.

**Exemple :**

- Nombres différents (3, 5) → Résultat : 8
- Nombres identiques (4, 4) → Résultat : 24 (triple de 8)

**Objectif :** Combiner conditions et opérations arithmétiques.

---

## Challenge 4 : Résolution d'Équation du Second Degré

Développez un programme qui résout une équation du second degré de la forme

$$ax^2 + bx + c = 0.$$

**Étapes à implémenter :**

1. Calculer le discriminant :  $\Delta = b^2 - 4ac$
2. Selon la valeur du discriminant :
  - $\Delta > 0$  : Deux solutions réelles distinctes
  - $\Delta = 0$  : Une solution double
  - $\Delta < 0$  : Pas de solution réelle

**Objectif :** Appliquer des conditions complexes et des calculs mathématiques avancés.

---

## Challenge 5 : Convertisseur d'Années (Menu Interactif)

Créez un programme avec menu qui convertit une année donnée en différentes unités temporelles.

**Options du menu :**

1. Conversion en mois
2. Conversion en jours
3. Conversion en heures
4. Conversion en minutes
5. Conversion en secondes

**Approximations :** 1 an = 365 jours, 1 mois = 30 jours

**Objectif :** Implémenter un menu interactif avec `switch case` et effectuer des conversions d'unités.

---

## Challenge 6 : Classification de Nombres

Écrivez un programme qui détermine si un nombre est positif, négatif ou nul.

**Règles :**

- `nombre > 0` : Positif
- `nombre < 0` : Négatif
- `nombre == 0` : Nul

**Objectif :** Maîtriser les opérateurs de comparaison et les conditions mutuellement exclusives.

---

## Challenge 7 : Vérification d'Alphabet Majuscule

Développez un programme qui vérifie si un caractère est une lettre majuscule en utilisant les valeurs ASCII.

**Information technique :**

- Lettres majuscules : A-Z
- Valeurs ASCII : 65 (A) à 90 (Z)
- Condition : `65 <= caractère <= 90`

**Objectif :** Comprendre les valeurs ASCII et les plages de caractères.

---

## Challenge 8 : Système de Mentions Académiques

Créez un programme qui attribue une mention selon la moyenne d'un étudiant.

**Barème de notation :**

- Moyenne < 10 : Échec (Recalé)
- $10 \leq$  Moyenne < 12 : Passable
- $12 \leq$  Moyenne < 14 : Assez Bien
- $14 \leq$  Moyenne < 16 : Bien
- Moyenne  $\geq$  16 : Très Bien

**Objectif :** Implémenter des conditions en cascade avec des plages de valeurs.

---

## Challenge 9 : Analyse Complète de Caractères

Écrivez un programme qui analyse un caractère et détermine :

1. S'il s'agit d'une lettre alphabétique
2. Si c'est une lettre, préciser si elle est majuscule ou minuscule

**Plages ASCII utiles :**

- Majuscules : 65-90 (A-Z)
- Minuscules : 97-122 (a-z)

**Objectif** : Combiner plusieurs conditions et classifications de caractères.

---

## Challenge 10 : Reformatage de Dates

Développez un programme qui convertit une date du format `JJ/MM/AAAA` vers le format `JJ-Mois-AAAA`.

**Exemple** : `15/09/2012` → `15-Septembre-2012`

**Astuce** : Utilisez un tableau de chaînes pour les noms des mois ou une structure `switch case`.

**Objectif** : Manipuler les chaînes de caractères et effectuer des conversions de format.

---

## Challenge 11 : Générateur de Jour Aléatoire

Créez un programme qui affiche aléatoirement un jour de la semaine.

**Approche** :

1. Générer un nombre aléatoire entre 1 et 7
2. Associer chaque nombre à un jour via `switch case`

**Note** : Utilisez `srand()` et `rand()` pour la génération aléatoire.

**Objectif** : Intégrer les fonctions aléatoires avec les structures conditionnelles.

---

## Challenge 12 : Comparateur d'Heures

Écrivez un programme qui compare deux instants au format `HH:MM:SS` et détermine leur ordre chronologique.

**Résultats possibles** :

- Le premier instant précède le second
- Le second instant précède le premier
- Les deux instants sont identiques

**Approche** : Comparer successivement les heures, puis les minutes, puis les secondes.

**Objectif** : Implémenter une logique de comparaison complexe avec plusieurs critères.

---

## Challenge 13 : Point sur un Segment (Géométrie)

Développez un programme qui détermine si un point donné se trouve sur un segment défini par deux extrémités.

**Données d'entrée** :

- Coordonnées des extrémités du segment :  $(x_1, y_1)$  et  $(x_2, y_2)$
- Coordonnées du point à tester :  $(x, y)$

**Condition mathématique** : Un point est sur un segment si :

1. Il est sur la droite qui contient le segment
2. Il se trouve entre les deux extrémités

**Objectif** : Appliquer des concepts géométriques et des conditions mathématiques complexes.