

# Exercices

---

Ces exercices vous donneront des exemples d'algorithmes répondant à une problématique donnée. L'ensemble de ces algorithmes peut former une base de révision algorithmique car ce document couvre les éléments nécessaires pour construire la plupart des algorithmes procéduraux existant.

## Problèmes

---

Vous trouverez la liste des problèmes et une de leurs solutions algorithmiques ci-dessous.

### 1. Echanger les valeurs de 2 variables

Problème : Echanger les valeurs de 2 variables R et Q

```
R <- 5
Q <- 34
```

Algorithme :

```
A <- R
R <- Q
Q <- A
```

### 2. Echanger les valeurs de 3 variables

Problème : Echanger les valeurs de 3 variables R, Q et V. Dans V, mettre Q, dans Q, mettre R et dans R, mettre V.

```
R <- 5
Q <- 34
V <- 45
```

Algorithme :

```
A <- V
V <- Q
Q <- R
R <- A
```

### 3. Vérifier si un utilisateur est majeur

Problème : Vérifier si un utilisateur est majeur grâce à son âge. Mettre une variable `resultat` à `Vrai` si cela est le cas, sinon mettez la à `Faux`.

```
age <- 25
```

Algorithme :

```
resultat <- Faux
@Si age >= 18
  @DebutBloc
    resultat <- Vrai
  @FinBloc
```

### 4. Vérifier si un utilisateur est mineur

Problème : Vérifier si un utilisateur est mineur grâce à son âge. Mettre une variable `resultat` à `Vrai` si cela est le cas, sinon mettez la à `Faux`.

```
age <- 12
```

Algorithme :

```
resultat <- Faux
@Si age < 18
  @DebutBloc
  resultat <- Vrai
  @FinBloc
```

ou bien

```
resultat <- Vrai
@Si age >= 18
  @DebutBloc
  resultat <- Faux
  @FinBloc
```

## 5. Echanger les 2 premiers éléments d'un tableau

Problème : Echanger les 2 premiers éléments d'un tableau

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

```
A <- tab[0]
tab[0] <- tab[1]
tab[1] <- A
```

## 6. Compter le nombre d'éléments dans un tableau

Problème : Compter le nombre d'éléments dans un tableau. Mettre le compte dans `resultat` .

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 7. Faire la somme des éléments d'un tableau

Problème : Faire la somme des éléments d'un tableau

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 8. Trouver l'élément maximum d'un tableau

Problème : Trouver l'élément maximum d'un tableau comprenant des nombres de 0 à 1000. Mettre le nombre maximum dans `resultat` .

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 9. Trouver l'élément minimum d'un tableau

Problème : Trouver l'élément minimum d'un tableau comprenant des nombres de 0 à 1000

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 10. Trouver le premier élément supérieur à 500

Problème : Trouver le premier élément supérieur à 500. S'il n'y en a pas, le resultat doit être 0

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 11. Copier un tableau

Problème : Copier le tableau `tab` dans un autre tableau `autre`

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

## 12. Copier les premiers éléments d'un tableau dont la somme fait au moins 500

Problème : Copier les premiers éléments d'un tableau dont la somme fait au moins 500

```
tab <- [23, 4, 2, 543, 34]
```

Algorithme :

Ou bine :