Exercices

Ces exercices vous donneront des exemples d'algorithmes répondant à une problématiques données. 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 procéduriaux 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
```

Javascript :

```
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
```

Javascript :

```
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
```

Javascript :

```
let resultat = false;
if (age >= 18)
{
    resultat = true;
}
```

ou bien

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

Javascript :

```
let resultat = null;
if (age < 18)
{
    resultat = false;
}
if (age >= 18)
{
    resultat = true;
}
```

ou bien

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

Javascript :

```
let resultat = true;
if (age < 18)
{
    resultat = false;
}
```

ou bien

```
@Si age >= 0 @Et age < 18
    @DebutBloc
    resultat <- Faux
    @FinBloc
@Si age >= 18
    @DebutBloc
    resultat <- Vrai
    @FinBloc
```

Javascript :

```
let resultat = null;
if (age >= 0 && age < 18)
{
    resultat = false;
}
if (age >= 18)
{
    resultat = true;
}
```

ou bien

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

Javascript :

```
let resultat = false;
if (!(age < 18))
{
    resultat = true;
}
```

ou bien

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

Javascript :

```
let resultat = null;
if (age < 18)
{
    resultat = false;
}
else
{
    resultat = true;
}
```

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
```

Javascript :

```
let resultat = false;
if (age < 18)
{
    resultat = true;
}
```

ou bien

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

Javascript :

```
let resultat = true;
if (age >= 18)
{
    resultat = false;
}
```

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
```

Javascript :

```
let 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 :

```
resultat <- 0
@PourChaque a @Dans tab
    @DebutBloc
    resultat <- resultat + 1
    @FinBloc
```

Javascript :

```
let resultat;
resultat = 0;
for (let a of tab)
{
    resultat = resulat + 1;
}
```

7. Faire la somme des éléments d'un tableau

Problème : Faire la somme des éléments d'un tableau. Mettre la somme dans une variable `resultat`.

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

Algorithme :

```
resultat <- 0
@PourChaque a @Dans tab
    @DebutBloc
    resultat <- resultat + a
    @FinBloc
```

Javascript :

```
let resultat = 0;
for (const a of tab)
{
    resultat = resultat + a;
}
```

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 numbre maximum dans `resultat`.

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

Algorithme :

```
la_somme_plus_elevee <- 0
@PourChaque h @Dans tab
    @DebutBloc
    @Si h > la_somme_plus_elevee
        @DebutBloc
        la_somme_plus_elevee <- h
        @FinBloc
    @FinBloc
resultat <- la_somme_plus_elevee
```

Javascript :

```
let la_somme_plus_elevee = 0;
for (const h of tab)
{
    if (h > la_somme_plus_elevee)
    {
        la_somme_plus_elevee = h;
    }
}
let resultat = la_somme_plus_elevee;
```

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 :

```
resultat <- 1000
@PourChaque h @Dans tab
    @DebutBloc
    @Si h < resultat
        @DebutBloc
        resultat <- h
        @FinBloc
    @FinBloc
```

Javascript :

```
let resultat = 1000;
for (const h of tab)
{
    if (h < resultat)
    {
        resultat = h;
    }
}
```

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 :

```
resultat <- 0
@PourChaque a @Dans tab
    @DebutBloc
    @Si a >= 500
        @DebutBloc
        resultat <- a
        @Stop
        @FinBloc
    @FinBloc
```

Javascript :

```
let resultat = 0;
for (const a of tab)
{
    if (a >= 500)
    {
        resultat = a;
        break;
    }
}
```

11. Copier un tableau

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

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

Algorithme :

```
autre <- []
@PourChaque a @Dans tab
    @Debutbloc
    Ajouter a dans autre
    @FinBloc
```

Javascript :

```
let autre = [];
for (const a of tab)
{
    autre.push(a);
}
```

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 :

```
autre <- []
somme <- 0
@PourChaque a @Dans tab
    @DebutBloc
    somme <- somme + a
    Ajouter a dans autre
    @Si somme >= 500
        @DebutBloc
        @Stop
        @FinBloc
    @FinBloc
```

Javascript :

```
let autre = [];
let somme = 0;
for (const a of tab)
{
    somme = somme + a;
    autre.push(a);
    if (somme >= 500)
    {
        break;
    }
}
```