

# Fiche résumé Algorithmique - Exemples

## Exemple1 - définir les données nécessaires, calculs arithmétiques

Calculer puis afficher un montant TTC à partir d'un montant HT et d'un taux de TVA à 20%

```
1  Algorithme calculerTTC
2  Données
3  Constantes
4  TAUX_TVA : Réel ← 0.20      // taux de TVA au 1er janvier 2014
5  Variables
6  montantHT : Réel           // saisi
7  montantTVA : Réel          // calculé
8  montantTTC : Réel          // résultat calculé
9  Début traitement
10 |
11 .. Afficher « quel est le montant HT ? »
12 | Saisir montantHT
13 |
14 | montantTVA ← ( montantHT * TAUX_TVA )
15 | montantTTC ← ( montantHT + montantTVA )
16 |
17 | Afficher « le montant TTC est », montantTTC
18 |
19 Fin traitement
```

# Fiche résumé Algorithmique - Exemples

## Exemple2 - structure conditionnelle

Rédiger l'algorithme permettant de déterminer si un enfant peut être admis en crèche (il doit avoir moins de 3 ans). On demandera son année de naissance et on connaît l'année actuelle.

1	<i>Algorithme déterminerCreche</i>
2	<i>Données</i>
3	<i>Constantes</i>
4	<i>ANNEE : Entier ← 2014 // année courante (attention à modifier )</i>
5	<i>AGE_LIMITE : Entier ← 3 // limite d'age</i>
6	<i>Variables</i>
7	<i>anneeNaissance : Entier // saisi</i>
8	<i>age : Entier // calculé</i>
9	<i>estAdmis : Booléen // calculé</i>
10	
11	<i>Début traitement</i>
12	
13	.. <i>Afficher « quel est son année de naissance ? »</i>
14	<i>Saisir anneeNaissance</i>
15	
16	<i>age ← ( ANNEE - anneeNaissance )</i>
17	<i>estAdmis ← ( age &lt; AGE_LIMITE )</i>
18	
19	<i>Si ( estAdmis )</i>
20	<i>Alors</i>
21	<i>Afficher « oui, il peut être admis à », age, « ans »</i>
22	<i>Sinon</i>
23	<i>Afficher « non, il ne peut être admis à », age, « ans »</i>
24	<i>finSi</i>
25	
26	<i>Fin traitement</i>

### Solution plus simple :

1. Suppression des lignes 9 et 17
2. Remplacement de la ligne 19 par :

19 : *Si ( age < AGE\_LIMITE )*

# Fiche résumé Algorithmique - Exemples

## Exemple3 - structure itérative TantQue (avec condition de poursuite de la répétition)

On souhaite calculer la moyenne d'un certain nombre de notes : l'utilisateur donnera ce nombre de notes, puis on permettra la saisie de ces notes, la moyenne sera calculée puis affichée

```
1  Algorithme calculerMoyenne
2  Données
3
4  Variables
5   nbreNotes : Entier           // saisi
6   compteurNotes : Entier       // compteur calculé
7   note : Réel                 // saisi
8   sommeNotes : Réel           // calculé
9   moyenneNotes : Réel         // résultat calculé
10
11 Début traitement
12 |
13 | .. Afficher « combien de notes ? »
14 | Saisir nbreNotes
15 | compteurNotes ← 1
16 | sommeNotes ← 0
17 |
18 | TantQue ( compteurNotes <= nbreNotes )
19 | | Afficher « quelle est la note numéro », compteurNotes, « ? »
20 | | Saisir note
21 | | sommeNotes ← ( sommeNotes + note )
22 | | compteurNotes ← ( compteurNotes + 1 )
23 | finTantQue
24 |
25 | moyenneNotes ← ( sommeNotes / nbreNotes )
26 |
27 | Afficher « la moyenne est », moyenneNotes
28 |
29 Fin traitement
```

# Fiche résumé Algorithmique - Exemples

## Exemple4 - structure itérative Pour (on connaît le nombre d'itérations)

On souhaite calculer la moyenne d'un certain nombre de notes : l'utilisateur donnera ce nombre de notes, puis on permettra la saisie de ces notes, la moyenne sera calculée puis affichée

1	<i>Algorithme</i> calculerMoyenne
2	<i>Données</i>
3	<i>Variables</i>
4	<i>nbreNotes</i> : Entier           // saisi
5	<i>compteurNotes</i> : Entier       // compteur calculé
6	<i>note</i> : Réel                 // calculé
7	<i>sommeNotes</i> : Réel         // calculé
8	<i>moyenneNotes</i> : Réel       // résultat calculé
9	
10	<i>Début traitement</i>
11	
12	.. <i>Afficher</i> « combien de notes ? »
13	<i>Saisir</i> <i>nbreNotes</i>
14	<i>sommeNotes</i> ← 0
15	
16	<i>Pour</i> <i>compteurNotes</i> <i>De</i> 1 à <i>nbreNotes</i> <i>PAS DE</i> 1
17	<i>Afficher</i> « quelle est la note numéro », <i>compteurNotes</i> , « ? »
18	<i>Saisir</i> <i>note</i>
19	<i>sommeNotes</i> ← ( <i>sommeNotes</i> + <i>note</i> )
20	<i>finPour</i>
21	
22	<i>moyenneNotes</i> ← ( <i>sommeNotes</i> / <i>nbreNotes</i> )
23	
24	<i>Afficher</i> « la moyenne est », <i>moyenneNotes</i>
25	
26	<i>Fin traitement</i>

# Fiche résumé Algorithmique - Exemples

## Exemple5 - structure conditionnelle et itérative, et lire une source externe

On souhaite calculer le taux de réussite d'une classe en fonction de la moyenne. On calcule le taux de réussite de la manière suivante : nombre de notes supérieures ou égales à la moyenne divisé par le nombre totale de notes.

On dispose d'un tableau comportant les colonnes numéro (pour le numéro d'étudiant) et moyenne.

Proposer un algorithme qui va lire les lignes du tableau et affiche le taux de réussite.

```
1  Algorithme calculerReussite
2  Données
3  Variables
4  nbrePlus : Entier           // nombre au dessus de la moyenne
5  nbreTotal : Entier         // nombre de notes total
6  tauxReussite : Réel        // taux de réussite
7  numero : Entier           // lu
8  moyenne : Réel            // lu
9
10 Début traitement
11
12   nbrePlus ← 0
13   nbreTotal ← 0
14
15   Lire Premier numero, moyenne De tableau
16   TantQue ( numero <> 0 )
17   | Si ( moyenne >= 10 )
18   | | Alors
19   | | | nbrePlus ← ( nbrePlus + 1 )
20   | | finSi
21   | | nbreTotal ← ( nbreTotal + 1 )
22   | Lire Suivant numero, moyenne De tableau
23   finTantQue
24
25   Si ( nbreTotal > 0 )
26   | Alors
27   | | tauxReussite ← ( nbrePlus / nbreTotal )
28   | | Afficher « le taux de succès vaut », tauxReussite
29   | Sinon
30   | | Afficher « pas de notes lues ! »
31   | finSi
32 Fin traitement
```

# Fiche résumé Algorithmique - Exemples

## Exemple6 - structure itérative et conditionnelle, et utilisation d'une fonction

On souhaite calculer la somme des montants HT des factures qui sont soumises à une taxe. La colonne 1 contient le numéro de facture, la colonne 5 est à 1 si la ligne est soumise à TVA, et la colonne 8 contient le montant HT à prendre en compte. Le détail des factures commence à la ligne 10 et s'arrête avec un numéro de facture à 0.

Une fonction **lireFacture** donne accès aux données qui se trouvent dans le tableau. Elle attend un numéro de ligne et un numéro de colonne et renvoie la valeur correspondante.

Proposer l'algorithme qui va permettre d'exploiter les données et afficher le montant TTC à partir de la somme des montants HT et d'un taux de TVA à 20%

1	<i>Algorithme calculerTTCFactures</i>
2	<i>Données</i>
3	<i>Constantes</i>
4	<i>TAUX_TVA : Réel ← 0.20 // taux de TVA au 1<sup>er</sup> janvier 2014</i>
5	<i>LIGNE_DEBUT : Entier ← 10 // ligne début des factures</i>
6	<i>COLONNE_NUM : Entier ← 1 // colonne des numéros</i>
7	<i>COLONNE_TAX : Entier ← 5 // colonne des taxes</i>
8	<i>COLONNE_MT : Entier ← 8 // colonne des montants</i>
9	<i>Variables</i>
10	<i>numeroFacture : Entier // récupéré</i>
11	<i>avecTaxe : Entier // récupéré</i>
12	<i>montantHT : Réel // récupéré</i>
13	<i>sommeMontantsHT : Réel // calculé</i>
14	<i>montantTotalTTC : Réel // résultat calculé</i>
15	<i>numeroLigne : Entier // numero de la ligne courante</i>
16	
17	<i>Début traitement</i>
18	
19	<i>numeroLigne ← LIGNE_DEBUT</i>
20	<i>sommeMontantsHT ← 0</i>
21	
22	<i>TantQue ( lireFacture(numeroLigne, COLONNE_NUM) &lt;&gt; 0)</i>
23	<i>Si (lireFacture(numeroLigne, COLONNE_TAX) = 1)</i>
24	<i>Alors</i>
25	<i>sommeMontantsHT ← sommeMontantsHT</i>
26	<i>+ lireFacture(numeroLigne, COLONNE_MT)</i>
27	<i>finSi</i>
28	<i>numeroLigne ← numeroLigne + 1</i>
29	<i>finTantQue</i>
30	
31	<i>montantTotalTTC ← (sommeMontantsHT * (1 + TAUX_TVA))</i>
32	<i>Afficher « le montant TTC total est », montantTotalTTC</i>
33	<i>Fin traitement</i>

# Fiche résumé Algorithmique - Exemples

## Exemple7 – fonction qui utilise une fonction

On souhaite calculer le taux de réussite d'une classe en fonction d'une limite passée à la fonction. On calcule le taux de réussite de la manière suivante : nombre de notes supérieures ou égales à la limite divisé par le nombre totale de notes.

On dispose d'une fonction **valeurCellule** qui attend un numéro de ligne et un numéro de colonne et qui retourne la valeur située dans cette cellule.

La plage à exploiter commence à la ligne 11 et se termine à la ligne 78. Le numéro d'étudiant est situé à la colonne 5, la moyenne à la colonne 8 et son nom à la colonne 2.

Proposer une fonction qui attend une note limite, va exploiter la feuille de calcul et retourner le taux de réussite.

1	<b>Fonction</b> calculerReussite (noteLimite : Réel) : Réel
2	<b>Données</b>
3	<b>Constantes</b>
4	LIGNEDEBUT : Entier ← 10
5	LIGNEFIN : Entier ← 78
6	COLONNEMOYENNE : Entier ← 8
7	<b>Variables</b>
8	nbrePlus : Entier                   // nombre au dessus de la moyenne
9	nbreTotal : Entier                 // nombre de notes total
10	tauxReussite : Réel                / taux de réussite
11	numeroLigne : Entier               // calculé, compteur
12	
13	<b>Début traitement</b>
14	nbrePlus ← 0
15	nbreTotal ← 0
16	
17	<b>Pour</b> numeroLigne <b>De</b> LIGNEDEBUT <b>A</b> LIGNEFIN <b>Pas de</b> 1
18	<b>Si</b> (valeurCellule(numeroLigne, COLONNEMOYENNE) >= NoteLimite)
19	<b>Alors</b>
20	nbrePlus ← ( nbrePlus + 1 )
21	<b>finSi</b>
22	nbreTotal ← ( nbreTotal + 1 )
23	<b>finPour</b>
24	
25	tauxReussite ← 0
26	<b>Si</b> (nbreTotal > 0)
27	<b>Alors</b>
28	tauxReussite ← ( nbrePlus / nbreTotal )
29	<b>finSi</b>
30	<b>Retourner</b> tauxReussite
31	<b>Fin traitement</b>

# Fiche résumé Algorithmique - Exemples

## Exemple8 - fonction et paramètres Plage

On souhaite calculer le taux de réussite d'une classe en fonction de la moyenne. On calcule le taux de réussite de la manière suivante : nombre de notes supérieures ou égales à la moyenne divisé par le nombre totale de notes.

On dispose d'une liste (provenant d'une plage de valeurs) comportant les colonnes numéro (pour le numéro d'étudiant) et moyenne.

Proposer une fonction qui va lire les lignes de la liste et retourner le taux de réussite.

```
1  Fonction calculerReussite (liste : Plage) : Réel
2  Données
3  Variables
4  nbrePlus : Entier // nombre au dessus de la moyenne
5  nbreTotal : Entier // nombre de notes total
6  tauxReussite : Réel / taux de réussite
7  numero : Entier // lu
8  moyenne : Réel // lu
9
10 Début traitement
11
12 nbrePlus ← 0
13 nbreTotal ← 0
14
15 Lire Premier numero, moyenne De liste
16 TantQue ( numero <> 0 )
17   Si (moyenne >= 10 )
18     Alors
19       nbrePlus ← ( nbrePlus + 1 )
20     finSi
21     nbreTotal ← (nbreTotal + 1 )
22     Lire Suivant numero, moyenne De liste
23   finTantQue
24
25   Si (nbreTotal > 0)
26     Alors
27       tauxReussite ← ( nbrePlus / nbreTotal )
28     Sinon
29       tauxReussite ← 0
30     finSi
31   Retourner tauxReussite
32 Fin traitement
```