

# Priorité des opérations

## Construction de l'algorithme PEMDAS<sup>1</sup> par les élèves

### Proposition

Dans ce document, il est proposé de faire redécouvrir progressivement les règles de la priorité des opérations par les élèves et de leur demander ensuite de construire l'algorithme « PEMDAS ».

### Champ d'action

Il est important de bien définir les conditions dans lesquelles nous allons fonctionner pour cet exercice. Au niveau des expressions mathématiques, nous posons les bases suivantes :

- Les expressions ne contiendront que des nombres, aucune lettre ;
- Les expressions contiendront les opérateurs +, -, x, : et exposant ;
- Les expressions pourront contenir de 0 à un nombre fini de paires de parenthèses ;
- Les expressions ne contiendront pas de parenthèses imbriquées (crochets, accolades).

### Objectif

Faire (re)découvrir par les élèves les règles PEMDAS et leur faire construire cet algorithme sur le canevas de ceux des robots virtuels du site <https://fadagogo.com>

### Ruse pédagogique (comme on dit au Sénégal)

Donner aux élèves une série d'expressions mathématiques selon les critères ci-dessus avec les résultats corrects des calculs et leur demander de définir la manière de procéder au niveau de la priorité des opérations afin d'arriver aux bons résultats. La quantité des exemples est réduite, il serait certainement judicieux d'en présenter un plus grand nombre aux élèves.

## 1. MDAS

Commençons par des expressions contenant les opérateurs +, -, x et : mais pas les exposants ni les parenthèses. Demandons aux élèves d'effectuer les calculs en utilisant toutes les combinaisons d'opérations possibles et d'ensuite comparer les résultats obtenus avec le résultat correct.

Demandons enfin de définir l'ordre correct des opérations à effectuer.

$$2 + 3 \times 5 = 17$$

$$5 \times 5 = 25 \rightarrow \text{Faux}$$

$$2 + 15 = 17 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer le x avant d'effectuer le +*

---

1 PEMDAS est un acronyme permettant de mémoriser la priorité des opérations, à savoir Parenthèses, Exposant, Multiplication, Division, Addition et Soustraction.

$$5 \times 4 - 8 : 2 = 16$$

$$20 - 8 : 2 = 12 : 2 = 6 \rightarrow \text{Faux}$$

$$5 \times -4 : 2 = -20 : 2 = -10 \rightarrow \text{Faux}$$

$$5 \times 4 - 4 = 5 \times 0 = 0 \rightarrow \text{Faux}$$

$$20 - 4 = 16 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer le x et le : avant d'effectuer le -*

À ce stade, nous pouvons déjà envisager de construire l'algorithme MDAS.

Nous voyons que plusieurs signes identiques peuvent figurer dans une même expression, l'utilisation d'une **répétitive** est donc indispensable.

Il existe deux types de répétitives, le **Répète** et le **Tant que**.

La vidéo [https://fadagogo.com/algorithmique\\_et\\_programmation/images\\_new/Les\\_repetitives.mp4](https://fadagogo.com/algorithmique_et_programmation/images_new/Les_repetitives.mp4) explique la différence entre ces deux répétitives. Il est judicieux de la regarder... :O)

Il se peut qu'un de ces 4 opérateurs ne se trouve pas dans l'expression. On utilisera donc le **Tant que** pour vérifier sa présence avant d'effectuer les opérations qui le concernent.

L'algorithme MDAS prendra donc la forme suivante :

**Tant que** Présence d'un signe x **Fais**

Réécris l'expression en exécutant le signe x

**Fin Tant Que**

**Tant que** Présence d'un signe : **Fais**

Réécris l'expression en exécutant le signe :

**Fin Tant Que**

**Tant que** Présence d'un signe + **Fais**

Réécris l'expression en exécutant le signe +

**Fin Tant Que**

**Tant que** Présence d'un signe - **Fais**

Réécris l'expression en exécutant le signe -

**Fin Tant Que**

Ceci doit permettre de trouver dans tous les cas le résultat final correct. À tester sur les expressions données en début d'exercice selon la technique du « double exécutant » (voir en fin de document).

## 2. EMDAS

Ajoutons maintenant des exposants dans les expressions abordées.

$$2 + 2^3 = 10$$

$$4^3 = 64 \rightarrow \text{Faux}$$

$$2 + 8 = 10 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'exposant avant d'effectuer le +*

$$3 \times 2^2 = 12$$

$$6^2 = 36 \rightarrow \text{Faux}$$

$$3 \times 4 = 12 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'exposant avant d'effectuer le x*

Nous pouvons maintenant réaliser l'algorithme EMDAS en complétant l'algorithme MDAS. Du fait de la priorité de l'exposant sur l'algorithme MDAS, la portion de code supplémentaire se trouvera avant l'algorithme MDAS afin d'être traitée en premier (principe de la séquence).

**Tant que** Présence d'un exposant **Fais**

Réécrit l'expression en exécutant l'exposant

**Fin Tant Que**

+ **Algorithme MDAS**

### 3. PEMDAS

Introduisons maintenant les parenthèses. Remarque : il est possible d'avoir une ou plusieurs paires de parenthèses, mais elles ne seront jamais imbriquées (conditions de départ).

$$2 \times (3 + 4) = 14$$

$$6 + 4 = 10 \rightarrow \text{Faux}$$

$$8 + 3 = 11 \rightarrow \text{Faux}$$

$$2 \times 7 = 14 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'intérieur des parenthèses (selon EMDAS) avant d'effectuer le x*

$$14 - (4 : 2) = 12$$

$$10 : 2 = 5 \rightarrow \text{Faux}$$

$$7 - 4 = 3 \rightarrow \text{Faux}$$

$$14 - 2 = 12 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'intérieur des parenthèses (selon EMDAS) avant d'effectuer le -*

$$12 - (2^2 : 4) - (18 - 9 : 3^2) = -6$$

$$8 : 4 - (9 : 3^2) = 2 - (9 : 9) = 2 - 1 = 1 \rightarrow \text{Faux}$$

Nombreuses variantes...

$$12 - (4 : 4) - (18 - 9 : 9) = 12 - 1 - (18 - 1) = 12 - 1 - 17 = -6 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'intérieur des parenthèses (selon EMDAS) avant d'effectuer les -*

$$4 + 2 \times 4 - (6 - 2 \times 5)^2 = -4$$

$$6 \times 4 - (4 \times 5)^2 = 24 - (20)^2 = 24 - 400 = -376 \rightarrow \text{Faux}$$

Nombreuses variantes...

$$4 + 8 - (6 - 10)^2 = 12 - (-4)^2 = 12 - 16 = -4 \rightarrow \text{Correct}$$

*Il faut d'abord effectuer l'intérieur des parenthèses (selon EMDAS) avant d'effectuer l'exposant des parenthèses*

Nous pouvons maintenant nous intéresser à l'algorithme final, celui de PEMDAS.

Cet algorithme précédera celui de EMDAS mais il faudra aussi appliquer EMDAS à l'intérieur des parenthèses, sachant qu'il peut y avoir de 0 à plusieurs paires de parenthèses.

Un Tant Que sera donc nécessaire afin « d'épuiser » les paires de parenthèses avant de passer au EMDAS final. Il faudra aussi prévoir la suppression des parenthèses à la fin du traitement de chaque paire de parenthèses. Il ne faudra pas non plus oublier l'éventuelle présence d'un signe – devant la paire de parenthèses. L'ensemble de l'algorithme donnera donc :

**Tant que** Présence de parenthèses **Fais**

Choisis une paire de parenthèses précises que tu vas traiter

Applique l'algorithme **EMDAS** à l'intérieur de ces parenthèses

**Si** Les parenthèses sont précédées d'un signe – **Alors**

Supprime le signe – devant les parenthèses

Change le signe du nombre dans les parenthèses

**Fin Si**

Réécris l'expression en supprimant les parenthèses

**Fin Tant Que**

+ **Algorithme EMDAS**

Spontanément, nous en sommes arrivés à réutiliser des morceaux d'algorithmes simplement en les citant, nous avons ainsi abordé la notion d'annexes ou de **fonctions**. Il est parfois pratique de mettre certaines parties d'algorithmes en annexes et ensuite de les appeler quand c'est nécessaire, cela économise de la place. Mais ceci sort un peu de notre sujet.

In extenso, l'algorithme, presque final, PEMDAS sera le suivant (sauf erreur ou omission, merci de me signaler tout souci :O). Les indentations de textes sont importantes, elles indiquent la portée des **Tant que** et des **Si**.

1 **Tant que** Présence de parenthèses **Fais**

2       Choisis une paire de parenthèses précises que tu vas traiter

3       **Tant que** Présence d'un exposant **Fais**

4               Réécris l'expression en exécutant l'exposant

5       **Fin Tant Que** // (de la ligne 3)

6       **Tant que** Présence d'un signe x **Fais**

7               Réécris l'expression en exécutant le signe x

8       **Fin Tant Que**

9       **Tant que** Présence d'un signe : **Fais**

10              Réécris l'expression en exécutant le signe :

11       **Fin Tant Que**

12       **Tant que** Présence d'un signe + **Fais**

13              Réécris l'expression en exécutant le signe +

14       **Fin Tant Que**

15 **Tant que** Présence d'un signe - **Fais**  
 16 Réécris l'expression en exécutant le signe -  
 17 **Fin Tant Que** // (de la ligne 15)  
 18 **Si** Les parenthèses sont précédées d'un signe – **Alors**  
 19 Supprime le signe – devant les parenthèses  
 20 Change le signe du nombre dans les parenthèses  
 21 **Fin Si**  
 22 Réécris l'expression en supprimant les parenthèses  
 23 **Fin Tant Que** // (de la ligne 1 concernant les parenthèses)  
 24 **Tant que** Présence d'un exposant **Fais**  
 25 Réécris l'expression en exécutant l'exposant  
 26 **Fin Tant Que**  
 27 **Tant que** Présence d'un signe x **Fais**  
 28 Réécris l'expression en exécutant le signe x  
 29 **Fin Tant Que**  
 30 **Tant que** Présence d'un signe : **Fais**  
 31 Réécris l'expression en exécutant le signe :  
 32 **Fin Tant Que**  
 33 **Tant que** Présence d'un signe + **Fais**  
 34 Réécris l'expression en exécutant le signe +  
 35 **Fin Tant Que**  
 36 **Tant que** Présence d'un signe - **Fais**  
 37 Réécris l'expression en exécutant le signe -  
 38 **Fin Tant Que**

### Fin de l'algorithme.

Dans cet algorithme, nous constatons que les contenus des lignes 3 à 17 et des lignes 24 à 38 sont totalement identiques. Nous allons donc mettre ces deux sections dans une **fonction** unique et appeler cette fonction à partir des 2 endroits adéquats. L'algorithme, final cette fois, est le suivant :

1 **fonction\_PEMDAS** // (Définition de la fonction)  
 2 **Tant que** Présence d'un exposant **Fais**  
 3 Réécris l'expression en exécutant l'exposant  
 4 **Fin Tant Que**  
 5 **Tant que** Présence d'un signe x **Fais**  
 6 Réécris l'expression en exécutant le signe x  
 7 **Fin Tant Que**  
 8 **Tant que** Présence d'un signe : **Fais**  
 9 Réécris l'expression en exécutant le signe :  
 10 **Fin Tant Que**

11 **Tant que** Présence d'un signe + **Fais**  
 12 Réécris l'expression en exécutant le signe +  
 13 **Fin Tant Que**  
 14 **Tant que** Présence d'un signe - **Fais**  
 15 Réécris l'expression en exécutant le signe -  
 16 **Fin Tant Que**  
 17 **Fin fonction\_PEMDAS**  
 18 **// (Début du programme principal)**  
 19 **Tant que** Présence de parenthèses **Fais**  
 20 Choisis une paire de parenthèses précises que tu vas traiter  
 21 **fonction\_PEMDAS** // appel à l'exécution de cette fonction  
 22 **Si** Les parenthèses sont précédées d'un signe – **Alors**  
 23 Supprime le signe – devant ces parenthèses précises  
 24 Change le signe du nombre dans ces parenthèses précises  
 25 **Fin Si**  
 26 Réécris l'expression en supprimant ces parenthèses précises  
 27 **Fin Tant Que** // (de la ligne 19 concernant les parenthèses)  
 28 **fonction\_PEMDAS** // appel à l'exécution de cette fonction  
**Fin de l'algorithme.**

L'exécution de l'algorithme commence au début de programme principal, c'est-à-dire à la ligne 19. Ça peut sembler long et contraignant, mais il s'agit de tous les stades par lesquels notre raisonnement doit passer pour appliquer correctement cette consigne. Ce détail et cette rigueur peuvent permettre à certains élèves d'améliorer la bonne exécution de cette consigne. Gérer le traitement d'une consigne et l'exécuter concrètement sont deux choses différentes...

## La technique du double exécutant

Cela n'est pas évident mais, quand on exécute une consigne, un algorithme, une marche à suivre,... on doit faire preuve d'un dédoublement de personnalité.

Une partie de nous va gérer la marche à suivre, va interroger « la partie externe » sur le résultat des vérifications (surlignées en jaune) et effectuer les sauts de lignes en fonction de ces résultats (ligne suivante si le résultat est Vrai, en général saut de plusieurs lignes avant ou après si le résultat est Faux). On peut parler d'*exécution interne, automatique* ou *aveugle*.

Au moment où la ligne examinée correspond à une instruction (en caractères normaux), une autre partie de nous va s'appliquer à exécuter cette instruction concrètement sur le papier, en déplaçant un objet,... On peut parler d'*exécution externe, effective*.

Pour faire prendre conscience aux élèves de ce dédoublement de personnalité, on peut procéder à l'exercice suivant :

Un élève, *l'externe*, est devant le tableau avec l'exercice mathématique à réaliser et la craie. Il va répondre aux demandes de l'autre élève, *l'interne*, qui est face à la marche à suivre *mais ne voit absolument pas l'exercice ni ce que fait l'élève externe*. On peut faire cela en inscrivant la marche à suivre sur un tableau annexe en faisant en sorte que l'élève *interne* tourne le dos à l'exercice et à ce que fait l'élève *externe*.

1. L'élève *interne* lit la ligne du programme à voix haute (la première si c'est le début).
2. Si la ligne est une instruction,
  1. L'élève *interne* demande à l'élève *externe* de réaliser cette instruction concrètement ;
  2. L'élève *interne* passe ensuite à la ligne qui suit immédiatement la ligne qui vient d'être exécutée.
3. Si la ligne est une vérification (ligne commençant par **Si**, **Jusqu'à ce que** ou **Tant que**),
  1. L'élève *interne* énonce la vérification à l'élève *externe* et lui demande si cette vérification donne un résultat Vrai ou Faux.
  2. En fonction du résultat, l'élève *interne* saute à la ligne adéquate à exécuter dans la marche à suivre (voir ci-dessous).
4. Si la ligne est un **Fin tant que** :
  1. L'élève *interne* saute automatiquement au *Tant que* correspondant
5. Et le circuit recommence au point 1 jusqu'à la fin du programme ou marche à suivre.

Un tel exercice a un avantage énorme, celui de montrer que l'exécution d'une consigne se joue à deux niveaux. Il y a le « traitement » proprement dit du texte de la consigne qui est **immuable** et qui n'a rien à voir avec ce qu'on est en train de couvrir comme sujet, qu'il s'agisse de mathématiques, de grammaire... Dans toute consigne, il y a donc une technique de traitement automatique à adopter, l'algorithmique, qui est totalement indépendante du monde extérieur et qui est valable quelle que soit la consigne et le domaine qu'on est en train de couvrir.

Le traitement extérieur ne sera, lui, fonction que des instructions comprises dans la consigne et des réponses apportées aux vérifications de la consigne.

En tout état de cause, seront immuables les actions suivantes :

- La ligne à laquelle il faudra sauter en fonction du résultat pour un **Si**
  - Vrai : à la ligne suivante
  - Faux : à la ligne qui suit le *Sinon* (s'il existe) ou le *Fin Si* (s'il n'y a pas de *Sinon*)
- La ligne à laquelle il faudra sauter en fonction du résultat pour un **Jusqu'à ce que**
  - Vrai : à la ligne suivante
  - Faux : à la ligne qui suit le *Répète* correspondant
- La ligne à laquelle il faudra sauter en fonction du résultat pour un **Tant que**
  - Vrai : à la ligne suivante
  - Faux : à la ligne qui suit le *Fin tant que* correspondant
- Un **Fin tant que** provoquera un saut automatique à la ligne du *Tant que* correspondant.

Cette **distinction interne-externe** me paraît fondamentale pour faire comprendre la dissociation mentale à mettre en œuvre face à n'importe quelle consigne.

Bien à vous, bonne lecture et n'hésitez pas à me contacter pour tout renseignement complémentaire.



Rupert Meurice de Dormale

Site Internet : <https://fadagogo.com>

Courriel : [info@fadagogo.com](mailto:info@fadagogo.com)

Rédigé le 30 avril 2022. Dernière correction le 1<sup>er</sup> octobre 2022.