

DEVELOPPER UN ALGORITHME

Exercices

Sommaire

Partie 1	2
Exercice 1.1 : Calcul de la moyenne de 2 nombres	2
Exercice 1.2 : Calcul de l'aire et du volume d'une sphère	3
Exercice 1.3 : Calcul de la surface d'un secteur circulaire	3
Exercice 1.4 : Inversion de 2 valeurs	3
Exercice 1.5 : Calcul d'intérêts	3
Exercice 1.6 : Nombre Premier	3
Exercice 1.7 : Tri de nombres	3
Exercice 1.8 : Année Bissextile	4
Exercice 1.9 : Recherche des diviseurs d'un nombre	4
Exercice 1.10 : Calcul des nombres parfaits	4
Partie 2	5
Exercice 2.1 : Conversion Kilomètres <-> Miles	5
Exercice 2.2 : Conversion Degrés Fahrenheit (°F) <--> Degrés Celsius (°C)	6
Exercice 2.3 : Jeu de la fourchette	7
Exercice 2.4 : Barnabé fait ses courses	7
Exercice 2.5 : Jeu du 0 - 2	7
Exercice 2.6 : Ma bicyclette	7
Exercice 2.7 : Calcul du nombre de personnes	8
Partie 3	9
Exercice 3.1 : Rechercher un nombre dans un tableau	9
Exercice 3.2 : Rechercher une lettre dans un chaîne de caractères	9
Exercice 3.3 : Dénumérer les lettres de l'alphabet dans un tableau	9
Exercice 3.4 : Tri d'un tableau	9
Exercice 3.5 : Palindrome	9
Exercice 3.6 : Jeu du pendu	10
Exercice 3.7 : Recherche par dichotomie d'un élément dans un tableau classé	10
Exercice final : Calcul du PGCD de 2 entiers positifs (Codage uniquement)	11

Consignes

Cette série d'exercices peut être réalisée dans les contextes suivants :

1. Se familiariser avec les différentes notations algorithmiques.
2. Apprendre la syntaxe d'un langage de programmation (C++, C#, Java, Javascript, PHP, Python...)

Afin de réaliser ces exercices de mise en œuvre ;

Vous devez connaître les systèmes de numération et l'algèbre de Boole.

Vous vous aiderez des supports d'apprentissage « Algorithmie introduction » et « Algorithmie représentation ».

Au fil de votre avancement, vous apprendrez et comprendrez les bases de l'algorithmie :

- La notion de variable.
- Les structures de contrôle.
- Les structures itératives.
- Les tableaux.
- Les procédures et fonctions.
- Les paramètres et retour de fonctions.

Dans un premier temps

Vous écrirez les solutions dans le langage algorithmique (pseudocode **et** organigramme) puis les ferez valider par votre formateur.

Pour chaque solution, fournissez un jeu d'essai et vérifiez le bon comportement de vos algorithmes.

Dans un deuxième temps

Vous coderez ces solutions en utilisant le langage de programmation indiqué par votre formateur.

Partie 1

Exercice 1.1 : Calcul de la moyenne de 2 nombres

Exercice 1.1.1 :

L'utilisateur est invité à saisir 2 nombres entiers.

Le programme calcule la moyenne des 2 nombres et affiche le résultat sous forme de nombre réel double précision.

Exemple d'affichage en mode Console

Programme de calcul de la moyenne de 2 nombres.

Veuillez saisir 2 nombres séparés par un espace :

12 81

La moyenne de 12 et 81 est : 46.5

Appuyez sur une touche pour quitter.

Exercice 1.1.2 :

Après chaque calcul, l'utilisateur est invité à réaliser un autre calcul ou quitter le programme.

Exemple d'affichage en mode Console

Programme de calcul de la moyenne de 2 nombres.

Veuillez saisir 2 nombres séparés par un espace :

12 81

La moyenne de 12 et 81 est : 46.5

Souhaitez-vous faire un autre calcul ? (O/N)

O

Veuillez saisir 2 nombres séparés par un espace :

23 577

La moyenne de 23 et 577 est : 300

Souhaitez-vous faire un autre calcul ? (O/N)

N

Appuyez sur une touche pour quitter.

Exercice 1.1.3 :

L'utilisateur peut saisir autant de nombres entiers qu'il souhaite. Chaque nombre entier saisi doit être séparé par un espace. Le programme calcule la moyenne de tous les nombres saisis.

Exercice 1.2 : Calcul de l'aire et du volume d'une sphère

Lire le rayon **R** d'une sphère.

Calculer et afficher :

- Son aire = $4 \pi R^2$
- Son volume = $\frac{4}{3} \pi R^3$

Exercice 1.3 : Calcul de la surface d'un secteur circulaire

Lire le rayon **R** d'un cercle et un angle **A** (en degré(s)).

Calculer et afficher :

- L'aire du secteur circulaire = $\pi R^2 A / 360$

Exercice 1.4 : Inversion de 2 valeurs

Lire 2 nombres **a** et **b**.

Mettre le contenu de **a** dans **b** et celui de **b** dans **a**.

Exercice 1.5 : Calcul d'intérêts

Lire la somme initiale **S** placée sur un compte, l'intérêt **i** offert par la banque et le nombre **N** d'années de placement de la somme **S**.

Calculez la valeur acquise par la somme **S** placée pendant **N** années :

- avec un intérêt simple : $S (1 + N * i)$
- avec un intérêt composé : $S (1 + i)^N$

Exercice 1.6 : Nombre Premier

Lire un nombre **N** et déterminer s'il est un nombre premier.

Exercice 1.7 : Tri de nombres

Lire 2 nombres **a** et **b**. Les écrire dans l'ordre croissant.

Lire 3 nombres **a**, **b** et **c**. Les écrire dans l'ordre croissant.

Exercice 1.8 : Année Bissextile

Lire une année **A**.

Déterminer si l'année **A** est bissextile. Si **A** n'est pas divisible par 4, l'année n'est pas bissextile.

Si **A** est divisible par 4, l'année est bissextile sauf si **A** est divisible par 100 et pas par 400.

Ecrire un message « Bissextil » ou « Non bissextil » suivant le cas.

Exercice 1.9 : Recherche des diviseurs d'un nombre

Lire un nombre entier et déterminer tous ses diviseurs autres que 1 et lui-même.

Exercice 1.10 : Calcul des nombres parfaits

On souhaite écrire un programme de calcul des **N** premiers nombres parfaits. Un nombre est dit parfait s'il est égal à la somme de ses diviseurs, 1 compris.

Exemple :

$$6 = 1+2+3$$

6 est un nombre parfait.

L'algorithme retenu contiendra deux boucles imbriquées. Une boucle de comptage des nombres parfaits qui s'arrêtera lorsque le décompte sera atteint, la boucle interne ayant vocation à calculer tous les diviseurs du nombre examiné d'en faire la somme puis de tester l'égalité entre cette somme et le nombre.

Version 1 :

Ecrivez le programme complet qui affiche les N premiers nombres parfaits.

Exemple d'affichage en mode Console :

```
Programme de recherche des nombres parfaits.  
Combien de nombre parfaits souhaitez-vous afficher ?  
4
```

```
Affichage des 4 premiers nombres parfaits :
```

```
6    est un nombre parfait.  
28   est un nombre parfait.  
496  est un nombre parfait.  
8128 est un nombre parfait.
```

Version 2 :

Après chaque calcul, l'utilisateur est invité à réaliser un autre calcul ou quitter le programme.

Partie 2

Exercice 2.1 : Conversion Kilomètres <-> Miles

Exercice 2.1.1 :

L'utilisateur saisit une valeur en kilomètres comprise entre 0.01 et 1 000 000. Si la valeur est hors limite, l'utilisateur est invité à saisir une nouvelle valeur. Si la valeur est égale à "q", le programme se termine et se ferme.

Formule km vers mi :

$$1 \text{ miles} = 1.609 \text{ kilomètres}$$

Le programme affiche le résultat de la conversion sous forme de nombre réel double précision.

Exemple d'affichage en mode Console :

Programme de conversion Kilomètres -> Miles.

Saisir une valeur en Kilomètres :

17

17 Kilomètres = 10.5655 Miles

Appuyez sur une touche pour quitter.

Exercice 2.1.2 :

L'utilisateur peut choisir le sens de la conversion.

Il saisit une valeur à convertir avec son unité de mesure (km ou mi).

Si aucune unité de mesure n'est indiquée, le programme considère la valeur en kilomètres.

Exemple d'affichage en mode Console :

Programme de conversion Kilomètres <-> Miles.

Saisir une valeur et son unité de mesure (séparés par un espace) :

17 mi

17 Miles = 27.353 Kilomètres

Exercice 2.1.3 :

L'utilisateur peut saisir plusieurs valeurs à convertir.

Il saisit une valeur à convertir avec son unité de mesure (km ou mi).

Si aucune unité de mesure n'est indiquée, le programme considère la valeur en kilomètres.

A l'appui de la touche « Entrée », la valeur saisie est enregistrée dans un tableau en mémoire.

L'utilisateur peut enregistrer autant de valeurs qu'il souhaite.

Si l'utilisateur saisit la commande « go », le programme convertit toutes les valeurs mémorisées et affiche le résultat de chaque conversion.

La commande « quit » permet de quitter le programme. Tant que cette commande n'est pas saisie, l'utilisateur peut continuer à faire des conversions.

Exercice 2.2 : Conversion Degrés Fahrenheit (°F) <--> Degrés Celsius (°C)

Exercice 2.2.1 :

Soit "X" la valeur à convertir.

Formule °F vers °C : $^{\circ}\text{C} = (\text{X} - 32) * 5/9$

Formule °C vers °F : $^{\circ}\text{F} = (\text{X} * 9/5) + 32$

L'utilisateur saisit une valeur numérique comprise entre -459.67 et 5 000 000 suivi de l'unité de température :

- C pour Celsius
- F pour Fahrenheit

La valeur et l'unité de température sont séparés par un espace (exemple: 32 C pour 32 degrés Celsius). Si la valeur est hors limite, l'utilisateur recommence la saisie.

Pour information, le zéro absolu correspond à -459.67 Degrés Fahrenheit ou -273.15 degrés Celsius. En physique, rien ne peut être plus froid que le zéro absolu !

La température de la plus basse jamais enregistrée sur Terre est de -95 degrés Celsius.

Le programme affiche le résultat de la conversion sous forme de nombre réel double précision.

Exemple d'affichage en mode Console :

Programme de conversion Celsius <-> Fahrenheit.

Saisir une température avec son unité de mesure (C ou F) :

17 C

17 °C = 62.6 °F

Exercice 2.2.2 :

L'utilisateur saisit une unité de mesure (C ou F).

Il saisit ensuite une plage de valeurs (minimum, maximum).

Une fois les 2 valeurs saisies, le programme convertit toute la plage de valeur et affiche le résultat.

La commande « quit » permet de quitter le programme. Tant que cette commande n'est pas saisie, l'utilisateur peut continuer à faire des conversions.

Exemple d'affichage en mode Console :

Programme de conversion Celsius <-> Fahrenheit.

Saisir une unité de mesure (C ou F) :

C

Saisir la valeur minimum :

10

Saisir la valeur maximum (doit être supérieure à la valeur minimum) :

12

Résultat :

10 °C = 50 °F

11 °C = 51.8 °F

12 °C = 53.6 °F

Exercice 2.3 : Jeu de la fourchette

L'ordinateur « choisit » un nombre entier compris entre 0 et 100. Le joueur essaie de le deviner.

Lors de chaque essai, l'ordinateur affiche la « fourchette » dans laquelle se trouve le nombre qu'il a choisi.

Le choix du nombre par l'ordinateur sera simulé par génération d'un nombre aléatoire : **$N \leftarrow \text{RANDOM}(0,100)$**

Exercice 2.4 : Barnabé fait ses courses

Barnabé fait ses courses dans plusieurs magasins.

Dans chacun, il a dépensé 1 euro de plus que la moitié de ce qu'il avait en entrant.

Dans le dernier magasin, il dépense le solde.

Soit **S** la somme dont il disposait au départ (**S** > 1 euro).

Représenter l'algorithme permettant de trouver le nombre de magasins dans lesquels il a acheté.

Exercice 2.5 : Jeu du 0 - 2

A tour de rôle, l'ordinateur et le joueur choisissent un nombre qui ne peut prendre que 3 valeurs: **0, 1** ou **2**.

Le choix du nombre par l'ordinateur sera simulé par génération d'un nombre aléatoire : **$N \leftarrow \text{RANDOM}$**

Si la différence entre les nombres choisis vaut :

- **2** : le joueur qui a proposé le plus grand nombre gagne un point.
- **1** : le joueur qui a proposé le plus petit nombre gagne un point.
- **0** : aucun point n'est marqué.

Le jeu se termine quand un des deux joueurs (l'ordinateur ou le joueur humain) totalise 10 points ou quand l'être humain introduit un nombre négatif qui indique sa volonté d'arrêter de jouer.

Exercice 2.6 : Ma bicyclette

Réalisez l'algorithme et le programme correspondant au texte ci-dessous :

S'il fait beau demain, j'irai faire une balade à bicyclette.

Je n'ai pas utilisé ma bicyclette depuis plusieurs mois, peut-être n'est-elle plus en parfait état de fonctionnement.

Si c'est le cas, je passerai chez le garagiste avant la balade. J'espère que les réparations seront immédiates sinon je devrai renoncer à la balade en bicyclette. Comme je veux de toute façon profiter du beau temps, si mon vélo n'est pas utilisable, j'irai à pied jusqu'à l'étang pour cueillir les joncs.

S'il ne fait pas beau, je consacrerai ma journée à la lecture. J'aimerais relire le 1^{er} tome de Game of Thrones. Je pense posséder ce livre, il doit être dans la bibliothèque du salon. Si je ne le retrouve pas, j'irai l'emprunter à la bibliothèque municipale. Si le livre n'est pas disponible, j'emprunterai un roman policier. Je rentrerai ensuite directement à la maison.

Dès que j'aurai le livre qui me convient, je m'installerai confortablement dans un fauteuil et je me plongerai dans la lecture.

Exercice 2.7 : Calcul du nombre de personnes

Exercice 2.7.1 : Calculer le nombre de jeunes

Il s'agit de dénombrer toutes les personnes d'âge inférieur strictement à 20 ans parmi un échantillon de 20 personnes. L'utilisateur est invité à saisir les 20 valeurs.

Décrivez l'algorithme qui affiche le nombre de jeunes et codez la solution.

Exercice 2.7.2 : Afficher le nombre de personnes de chaque catégorie

Compléter l'algorithme précédent pour répondre à la demande suivante:

Si toutes les personnes ont moins de 20 ans, affichez « TOUTES LES PERSONNES ONT MOINS DE 20 ANS ».

Si aucune personne n'a moins de 20 ans, affichez « TOUTES LES PERSONNES ONT PLUS DE 20 ANS ».

Sinon, affichez le nombre de personnes pour chaque catégorie (« - de 20, + de 20, = à 20 »).

Jeu d'essai:

Pas de jeunes

45 35 65 76 34 32 31 46 57 68 75 46 53 36 31 46 68 59 30 20

Pas de non-jeunes

15 5 5 6 4 2 11 16 7 8 7 3 13 16 11 18 8 9 19 3

Des jeunes et des non-jeunes

45 35 65 76 34 20 20 30 30 30 20 20 30 20 30 20 20 8 15 23

Partie 3

Exercice 3.1 : Rechercher un nombre dans un tableau

Soit un tableau de nombres triés par ordre croissant.

Chercher si un nombre donné **N** figure parmi les éléments.

Si oui, indiquer la valeur de l'indice correspondant. Sinon, indiquer « Non trouvé ».

Exercice 3.2 : Rechercher une lettre dans un chaîne de caractères

Soit une chaîne de caractères terminée par le caractère « . ».

Donnez l'algorithme d'un programme qui compte le nombre d'occurrences d'une lettre donnée (« a » par exemple) dans cette chaîne.

Si la chaîne de caractères n'est composée que du caractère « . », le message « LA CHAINE EST VIDE » sera affiché.

Proposez un jeu d'essai prévoyant les 3 cas suivants :

- La phrase est vide
- La lettre n'est pas présente
- La lettre est présente plusieurs fois

Exercice 3.3 : Dénombrer les lettres de l'alphabet dans un tableau

Lire un texte d'au moins 120 caractères (à contrôler).

Compter et afficher le nombre d'occurrences (d'apparitions) de chacune des lettres de l'alphabet.

Exercice 3.4 : Tri d'un tableau

Nous désignerons par a_1, a_2, \dots, a_N les éléments d'un tableau à trier par ordre croissant.

On commence par chercher l'indice du plus petit des éléments, soit j cet indice.

On permute alors les valeurs de a_1 et a_j .

On cherche ensuite l'indice du plus petit des éléments a_2, a_3, \dots, a_n et on permute avec a_2 , etc.

Exercice 3.5 : Palindrome

Un palindrome est une chaîne de caractères que l'on peut lire identiquement de droite à gauche, et gauche à droite.

Par exemple:

AA. 38783. LAVAL. LAVAL A ETE A LAVAL. ET LA MARINE VA VENIR A MALTE

Soit une chaîne de caractères terminée par un point.

Ecrivez l'algorithme d'un programme permettant d'affirmer si cette phrase est ou non un palindrome.

Si la chaîne de caractères n'est composée que du caractère '.', le message 'LA CHAINE EST VIDE' sera affiché.

Proposez un jeu d'essai prévoyant les 3 cas suivants :

- la phrase est vide
- la chaîne de caractères n'est pas un palindrome
- la chaîne de caractères est un palindrome

Exercice 3.6 : Jeu du pendu

L'algorithme lit un mot proposé par un premier joueur.

Ce mot a une longueur minimum de 5 caractères (à contrôler).

L'algorithme affiche ensuite le mot où toutes les lettres sauf la première et la dernière sont remplacées par un tiret.

Un deuxième joueur propose des lettres une à une.

Chaque fois que la lettre se trouve dans le mot, l'algorithme remplace les tirets qui remplaçaient cette lettre et réaffiche le mot. Le second joueur a droit à un maximum de 6 essais pour retrouver toutes les lettres.

Exercice 3.7 : Recherche par dichotomie d'un élément dans un tableau classé

Recherche par dichotomie d'un élément dans une table classée

Soit une table contenant des prénoms, classés par ordre alphabétique

Nous désirons chercher l'indice de la case de la table où se trouve le prénom, si il s'y trouve.

Pour cela, nous utiliserons la méthode de dichotomie (voir ci-dessous la méthode)

Donnez l'algorithme de la procédure qui recherche, par dichotomie le numéro du prénom recherché ou zéro s'il n'y est pas.

Principe de la recherche par dichotomie:

Les prénoms sont classés par ordre alphabétique
On connaît le nombre d'éléments de la table

1 ->	agathe
2 ->	berthe
3 ->	chloé
4 ->	cunégonde
5 ->	olga
6 ->	raymonde
7 ->	sidonie

Algorithme:

On partitionne la table en 2 sous-tables et un élément médian, et, suivant le résultat de la comparaison de l'élément médian et du prénom recherché (plus grand, plus petit ou égal) on recommence la recherche sur une des 2 sous-tables, jusqu'à avoir trouvé ou obtenir une sous-table vide (le prénom est alors absent de la table).

On cherche 'olga' dans la table:

Milieu: élément 4

'olga' > 'cunégonde' -> 'olga' est entre 4 et 7

milieu: élément 6

'olga' < 'raymonde' -> 'olga' est entre 4 et 6

milieu: élément 5

'olga' trouvé en 5

Exercice final : Calcul du PGCD de 2 entiers positifs (Codage uniquement)

En arithmétique élémentaire, le plus grand commun diviseur ou PGCD de deux nombres entiers non nuls est le plus grand entier qui les divise simultanément.

Par exemple, le PGCD de 20 et de 30 est 10, puisque leurs diviseurs communs sont 1, 2, 5 et 10.

On souhaite écrire un programme de calcul du PGCD de deux entiers non nuls strictement positifs, à partir de l'algorithme de la méthode dite « égyptienne ».

Voici une spécification de l'algorithme de calcul du PGCD de deux nombres (entiers strictement positifs) p et q , selon cette méthode :

```
Lire (p, q)
  TantQue p ≠ q
  Faire
    Si p > q Alors
      p ← p - q
    Sinon
      q ← q - p
    FinSi
  Fin TantQue
Ecrire( " PGCD = ", p )
```

Version 1 :

Réalisez le programme correspondant. L'algorithme ci-dessus doit être implémenté dans une fonction nommée « Calcul_PGCD » qui accepte 2 nombres entiers en paramètres et retourne le PGCD calculé au format nombre entier. La signature de la méthode Calcul_PGCD est : **Calcul_PGCD (int a , int b) : int.**

Conservez votre fonction Calcul_PGCD, elle vous sera utile pour les exercices suivants.

Exemple d'affichage en mode Console :

Programme de calcul du PGCD de 2 nombres entiers positifs

Saisir le 1^{er} nombre :

176

Saisir le 2nd nombre :

892

Le PGCD de 176 et 892 est égal à 4.

Version 2 :

En vous inspirant des exercices précédents, proposez une version améliorée et intuitive du programme.

--- FIN DU DOCUMENT ---

<http://www.arfp.asso.fr>