CENTRE DE RÉADAPTATION MULHOUSE
Rééducation et Formation Professionnelle

Algorithmes Exercices #2

Exercices de logique

CONTENU

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 connaitre les systèmes de numération et l'algèbre de Boole.

Vous vous aiderez des supports d'apprentissage fournis par vos formateurs.

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/ou** 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 vos formateurs.

PARTIE 4

EXERCICE 4.1 : NOMBRE PREMIER

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

EXERCICE 4.2 : 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.

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.
```

Astuce: Lors de vos tests, afficher au maximum 4 nombres parfaits, tenter d'en afficher plus risque de prendre du temps 😊.

EXERCICE 4.3 : CONVERSION KILOMETRES <-> MILES

Exercice 4.3.1 :

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 miles = 1.609 kilomètres**

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

Exercice 4.3.2 :

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.

EXERCICE 4.4 : CONVERSION DEGRES FAHRENHEIT (°F) <--> DEGRES CELSIUS (°C)

Exercice 4.4.1 :

Soit "X" une valeur à convertir.

Formule °F vers °C : $\mathbf{°C = (X - 32)\ \frac{5}{9}}$

Formule °C vers °F : $\mathbf{°F = (X\ \frac{9}{5}) + 32}$

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 est invité à saisir une nouvelle valeur.

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

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 -95 degrés Celsius.

Exercice 4.4.2 :

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.

PARTIE 5

EXERCICE 5.1 : LES FRUITS ET LEGUMES

L'utilisateur peut saisir des noms de légumes. Pour chaque légume, l'utilisateur précise un prix au kilo.

Exemple : "carottes 2.99"

Lorsque l'utilisateur saisit la valeur "go", le programme affiche la liste des légumes avec leur prix ainsi que le légume le moins cher.

Exemple :

1 kilogramme de carottes coute 2.99 euros.
1 kilogramme de poireaux coute 1.99 euros.
etc…
Légume le moins cher au kilo : poireaux

EXERCICE 5.2 : 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 <-- RANDOM**

Si la différence entre les nombres choisi 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.

Dans les 2 cas, afficher les scores.

EXERCICE 5.3 : 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 1er 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 5.4 : CALCUL DU NOMBRE DE PERSONNES

Exercice 5.3.1 : Calculer le nombre de jeunes

Il s'agit de dénombrer toutes les personnes d'âge strictement inférieur 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 5.3.2 : 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

EXERCICE 5.5 : DENOMBRER LES LETTRES DE L'ALPHABET

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.

PARTIE 6

EXERCICE 6.1 : TRI D'UN TABLEAU

Soit T un tableau de nombres entiers non trié.
Trier le tableau et afficher tous ses éléments par ordre croissant.

On commence par chercher l'indice du plus petit des éléments, soit **j** cet indice.
On permute alors les valeurs de a1 et a**j** .
On cherche ensuite l'indice du plus petit des éléments a2, a3, ..., an et on permute avec a2, etc…

EXERCICE 6.2 : 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 un texte d'au moins 2 caractères (à contrôler).
Ecrivez l'algorithme d'un programme permettant d'affirmer si cette phrase est un palindrome.
Si la chaîne de caractères est vide, 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 6.3 : 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 proposée 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 erreurs pour retrouver toutes les lettres.

EXERCICE 6.4 : RECHERCHE PAR DICHOTOMIE D'UN ELEMENT DANS UN TABLEAU CLASSE

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 précédente :

Milieu: élément 4

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

milieu: élément 6

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

mileu: élément 5

'olga' trouvé en 5

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 peut-être utile...

Version 2 :

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

EXERCICE FINAL : YAOURTS

OBJECTIF

Vous êtes en charge de la réalisation d'un algorithme d'analyse d'une étude marketing sur l'emballage d'un nouveau yaourt bio issu de circuits courts.

Vous recevez à ce titre les résultats d'une étude où des consommateurs indiquent la couleur qu'ils préfèrent pour l'emballage.
Vous décidez de présenter les 2 couleurs les plus demandées à votre client.

CONSIGNES

Rédigez l'algorithme et créez le programme dans le langage indiqué par vote formateur.

Votre programme contiendra obligatoirement une fonction **App.run(results : Array): String**

Cette fonction sera évaluée dans un test unitaire.

DONNEES

ENTREE

Un tableau indexé contenant entre 500 et 5000 entrées.
Chaque élément du tableau correspond à la couleur choisie par une personne interrogée.
Ces éléments du tableau sont générés aléatoirement à chaque appel de votre algorithme.
On vous garantit qu'il n'y aura jamais 2 couleurs ex-aequo.

Exemple :

> ['rouge', 'jaune', 'bleu', 'jaune', 'rouge', 'jaune']

SORTIE

Une chaine de caractère représentant les deux couleurs préférées.
Les 2 valeurs sont séparées par un espace.
La couleur la plus populaire doit apparaître en premier.

Exemple :

Pour le tableau en entrée :

> ['rouge', 'jaune', 'bleu', 'jaune', 'rouge', 'jaune']

La réponse est :

> jaune rouge

Car le jaune a obtenu 3 votes, le rouge 2 votes et le bleu 1 vote.

--- FIN DU DOCUMENT ---