Université Mohammed Seddik BENYAHIA – JIJEL –
Faculté des Sciences Exactes et Informatique **2025 / 2026**
Départements d'Informatique + Mathématiques

– Algorithmique – TD – Série N°2 – Les structures de contrôle

Exercice 1 : Si … Alors … Sinon … FSi

Q1) Écrire un algorithme qui détermine si un entier positif **N** est pair ou impair.

Q2) Écrire un algorithme qui prend en entrée deux entiers **A** et **B** et affiche leur minimum.

Q3) Écrire un algorithme qui permet de résoudre l'équation du second degré : $\mathbf{aX^2 + bX + c = 0}$.

Q4) Une librairie facture **5 DA** les dix premières photocopies, **4 DA** les vingt suivantes et **3 DA** au-delà.

Écrire un algorithme qui lit le nombre de photocopies effectuées **(N)** et qui affiche la facture correspondante.

Q5) Une **année bissextile** est une année comptant **366 jours** au lieu de **365 jours** pour une **année normale**.

C'est-à-dire une année comprenant un **29 février**. La prochaine année bissextile est **2024**.

Une année est bissextile si elle est divisible par 4 mais pas par 100 sauf si elle est multiple de 400.

Exemples : 2000 et 2008 sont des années bissextiles. 2006 et 2100 sont des années normales (non- bissextiles).

Écrire un algorithme permettant de vérifier si une année est bissextile ou pas.

Q6) Écrire un algorithme qui permet d'ordonner trois nombres entiers (**A, B, C**) dans l'ordre croissant.

Exercice 2 : Essentiellement la boucle POUR …

Ecrire des algorithmes pour les cas suivants :

Q1) Calcul de la **somme** $S = 1 + 2 + 3 + \ldots + N$.

Q2) Calcul de la **somme** $S = 2^2 + 4^2 + 6^2 + \ldots$ en prenant N termes.

Q3) Calcul de la **puissance** N d'un nombre réel X *i.e.* $X^N = X * X * \ldots * X$, N fois.

Q4) Calcul de la **factorielle** d'un entier naturel N i.e. $N ! = N * (N-1) * \ldots * 3 * 2 * 1$.

Q5) Calcul de la **somme** $S = 1! + 2! + 3! + \ldots + N!$

Q6) Calcul de la valeur approchée $e^x \cong 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$ (***x*** est un nombre réel, ***n*** un entier positif).

Q7) Calcul du ***sinus*** d'un angle ***x*** exprimé en radian est donné par la somme infinie suivante :

$$Sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \ldots$$

Q8) Ecrire un l'algorithme permettant de calculer le n^{ieme} terme de la suite de ***Fibonacci*** définie par :

$$\begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} \quad Si\ n \geq 2 \end{cases}$$

Exercice 3 : Nombres Parfaits

Q1) Ecrire un algorithme qui permet d'afficher tous les diviseurs d'un entier N.

Un nombre est dit **parfait** s'il est égal à la somme de tous ses diviseurs excepté lui-même.

Exemples : **6** est parfait car **6 = 1 + 2 + 3**. Les diviseurs de 6 sont : 1, 2, 3 et 6 (exclu).
28 est parfait car **28 = 1 + 2 + 4 + 7 + 14**. Les diviseurs de 28 sont : 1, 2, 4, 7, 14 et 28 (exclu).

Q2) Ecrire un algorithme qui permet de vérifier si un entier N est parfait ou pas.
Q3) Généraliser l'algorithme précédent pour afficher tous les nombres parfaits ≤ NMax.

Exercice 4 : PGCD et PPCM

Q1) L'algorithme d'Euclide permettant de calculer le **PGCD** (Plus Grand Commun Diviseur) de deux entiers strictement positifs $\boldsymbol{A}$ et $\boldsymbol{B}$ tel que $\boldsymbol{A \geq B}$ est défini comme suit :

$$PGCD(A,B) = \begin{cases} PGCD(B, A\,mod \quad B) & Si\ B \neq 0 \\ \qquad\quad A & Si\ B = 0 \end{cases}$$

Ecrire un algorithme qui permet de : a) Saisir deux entiers positifs non nuls $\boldsymbol{A}$ et $\boldsymbol{B}$. b) S'assurer que $\boldsymbol{A \geq B}$.

c) Déterminer et afficher le ***PGCD*** de $\boldsymbol{A}$ et $\boldsymbol{B}$.

Q2) Une méthode pour calculer le **PPCM** (Plus Petit Commun Multiple) de deux entiers strictement positifs $\boldsymbol{A}$ et $\boldsymbol{B}$ tel que $\boldsymbol{A \geq B}$ est de trouver le plus petit multiple de $\boldsymbol{A}$ qui est aussi multiple de $\boldsymbol{B}$.

Ecrire un algorithme permettant de trouver le PPCM de deux entiers positifs non nuls $\boldsymbol{A}$ et $\boldsymbol{B}$.

Exercice 5 : Nombre premier

Un nombre est dit **premier** s'il n'admet que deux diviseurs : **1** et lui-même.

Q1) Ecrire un algorithme qui permet de vérifier si un entier **N** est premier.

Q2) Modifier l'algorithme précédent pour afficher les vingt (20) petits nombres premiers.

Exercice 6 : Suite de Syracuse

A partir d'un entier **N** strictement positif **(N > 0)**; on construit la suite suivante :

- Si **N** est pair, on le remplace par **N/2**.
- Si **N** est impair, on le remplace par **3N+1**.

En répétant l'opération, on obtient une suite d'entiers positifs. On arrête la construction de cette suite, quand la valeur de **1** est rencontrée.

On appelle altitude maximale la valeur maximale de la suite obtenue.

- Si **N = 1**, la suite obtenue est : **1, 4, 2, 1**. **L'altitude maximale = 4.**
- Si **N = 13**, la suite obtenue est : **13, 40, 20, 10, 5, 16, 8, 4, 2, 1**. **L'altitude maximale = 40.**

Q1) Écrire un algorithme qui permet de : ①- Lire un entier **strictement positif**, ②- Afficher la suite obtenue ainsi que l'altitude maximale trouvée.

Q2) Compléter l'algorithme précédent pour trouver les **5** petits nombres qui ont une altitude maximale supérieure à une valeur donnée (**30** par exemple).

Exercice 7 : Nombres Symétriques

Soit **N** un nombre entier positif.

Q1) Ecrire un algorithme qui permet d'afficher les chiffres qui composent le nombre **N** ainsi que sa longueur.

Exemples : - Si **N = 17➔** on affiche les chiffres :7 puis **1** et **La longueur = 2**.
- Si **N = 695➔** on affiche les chiffres : **5** puis **9** puis **6** et **La longueur = 3**.

Q2) Ecrire un algorithme qui permet de calculer puis afficher **le nombre inverse de N**.

Exemple Si **N = 695➔** son nombre inverse = **596**.

Q3) Dérouler l'algorithme pour **N = 695**

Un nombre N est dit **symétrique** s'il est égal à son inverse.

Exemples : Les nombres suivants sont symétriques : **1, 2, 3, 44, 55, 161, 717, 8228, 94549.**

Q4) Modifier l'algorithme précédent pour qu'il affiche un message indiquant si N est symétrique ou non.

Q5) Généraliser l'algorithme précédent pour qu'il affiche tous les nombres symétriques de longueur égale à **5**.