

EXERCICE 1 (4 pts)

Ecrire un algorithme nommé **AFFICHE_SAISON** qui, à partir du numéro d'un mois saisi par l'utilisateur (1 pour janvier ; 2 pour février ; ... ; 12 pour décembre), permet d'afficher la saison.

Exemple :

Si mois = 6 ou 7 ou 8 Alors le programme affiche : **Eté**

Si mois = 3 ou 4 ou 5 Alors le programme affiche : **Printemps**

Si mois = 9 ou 10 ou 11 Alors le programme affiche : **Automne**

Si mois = 12 ou 1 ou 2 Alors le programme affiche : **Hiver**

EXERCICE 2 : (4 points)

Soit un module composé de 3 notes (comprises entre 0 et 20). Tout étudiant se trouvant dans au moins l'un des trois cas suivants sera refusé :

- deux des trois notes sont strictement inférieures à 7,5
- une note est strictement inférieure à 6
- la moyenne des trois notes est strictement inférieure à 10

Ecrire un algorithme qui saisit les trois notes N_1, N_2 et N_3 , puis annonce le résultat sous la forme "**admis**" ou "**refusé**".

EXERCICE 3 : (4 points)

Un magasin de reprographie facture à 50 francs la page, les dix premières photocopies, 25 francs la page, les vingt suivantes et 15 francs la page au-delà.

Ecrivez un algorithme qui demande à l'utilisateur le nombre de pages de photocopies à effectuer et qui affiche la facture correspondante.

EXERCICE 4 (4 pts)

Écrire un algorithme nommé **CalculSomme** qui demande un nombre N à l'utilisateur, et qui calcule la somme des entiers compris entre 1 (inclus) et N (inclus).

Par exemple : si l'utilisateur saisit le nombre 5, l'algorithme devra afficher :

La somme des 5 premier entiers est : 15

Remarque : tant que le nombre saisi est négatif ou nul, l'algorithme demande à l'utilisateur de saisir à nouveau le nombre.

Compléter l'algorithme suivant :



Devoir Surveillé d'Informatique

Algorithmique et Programmation Python

Durée: 1 heure 30 minutes

L'usage d'abaques, de tables, de calculatrice et de tout instrument électronique susceptible de permettre au candidat d'accéder à des données et de les traiter par les moyens autres que ceux fournis dans le sujet est interdit.

Chaque candidat est responsable de la vérification de son sujet d'épreuve : pagination et impression de chaque page. Ce contrôle doit être fait en début d'épreuve. En cas de doute, le candidat doit alerter au plus tôt le surveillant qui vérifiera et, éventuellement, remplacera le sujet.

Ce sujet comporte 2 pages numérotées de 1 à 2.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

Ce sujet porte sur l'étude de la croissance d'une population de lapins dans une situation idéale. On suppose la situation suivante: *quelqu'un a déposé un couple de lapins dans un endroit clos de tous côtés, pour savoir combien de couples seraient issus de cette paire en une année, car il est dans leur nature de générer un autre couple en un seul mois, et qu'ils enfantent dans le second mois après leur naissance.*

Ce problème est appelé problème de Fibonacci et est à l'origine de la suite de Fibonacci dont le n -ième terme correspond au nombre de paires de lapins au n -ième mois. Dans cette population idéale, on suppose que:

- au début du premier mois, il n'y a qu'une paire de lapereaux;
- les lapins ne peuvent procréer qu'à l'âge de deux mois;
- chaque début de mois, toute paire susceptible de procréer engendre exactement une nouvelle paire de lapereaux;
- les lapins ne meurent jamais.

1. Questions préliminaires

Notons F_n le nombre de couples de lapins au début du mois n .

- (a) Combien de couples de lapins a-t-on à la fin du premier mois (noté F_1)? De même quelles sont les valeurs de F_2 , F_3 , et F_4 ?

- (b) Plaçons-nous maintenant au mois n et cherchons à exprimer ce qu'il en sera deux mois plus tard, soit au mois $n+2$. On vous fait remarquer que les F_{n+2} couples de lapins sont formés des F_{n+1} couples du mois précédent et des couples nouvellement engendrés. Donnez l'expression de F_{n+2} .

2. Etude Algorithmique

L'étude préliminaire nous a permis de déterminer la forme récurrente de la fonction de la suite qui caractérise F_n (Section 1, Question b).

- Rédiger votre méthodologie de résolution algorithmique de ce problème en utilisant la méthode DRAST (seule les sections Données, Résultats et Actions doivent être mentionnées).
- Ecrire un algorithme en pseudo-code permettant de calculer F_n avec n qui sera précisé par l'utilisateur. Votre algorithme devra afficher la valeur de F_n à la fin de son exécution.
- Représenter graphiquement à l'aide d'un algogigramme votre algorithme.
- Ecrire un second algorithme en pseudo-code permettant toujours de calculer F_n avec n qui sera précisé par l'utilisateur. L'algorithme devra cependant afficher les valeurs successives prises par F_n (entre 0 et n) à la fin de son exécution.

3. Programmation Python

Dans la partie précédente, on implémente en pseudo-code un algorithme nous permettant de calculer F_n . On décide enfin de traduire notre algorithme en langage Python et de l'étendre à l'étude de p populations de lapins.

- Traduisez vos deux algorithmes trouvés dans la seconde partie (Section 2, questions a et b) en Python.
- Modifiez votre premier algorithme (Section 2, question a), traduit en Python, pour qu'il calcule F_n pour chaque population. Le nombre de population p est précisé par l'utilisateur et n est précisé pour chaque population. L'algorithme devra renvoyer un tableau de dimension 2 contenant pour chaque population le F_n correspondant.

FIN DU SUJET.

EXAMEN D'ALGORITHMIQUE**Exercice 1** (5 pts)

La Direction de la pédagogie de l'ESATIC veut saisir et afficher les notes des étudiants de la SRITI du premier semestre.

Ecrire un **Algorithme** qui permet la saisie du nombre de notes à saisir, puis des notes. On sait simplement que le nombre de notes ne peut dépasser **500**. L'algorithme affichera les notes sous la forme :

note numéro 1 : 18

note numéro 2 : 11

note numéro 3 : 13

etc.

ALGORITHME SaisieAffichageNotes

CONSTANTES

MAX_NB_NOTES = 500 /* le nombre maximal de notes */

VARIABLES

nbNotes : ENTIER /* nombre de notes à saisir */

notes : TABLEAU [1..MAX_NB_NOTES] DE REEL /* les notes */

i : ENTIER /* indice */

reponse : CHAINE /* réponse oui/non */

DEBUT

ECRIRE("Saisie des notes")

A COMPLETER

FIN

Exercice 2 (5 pts)

On considère qu'à l'ESATIC, une classe contient au plus 100 étudiants. Chaque étudiant est représenté par un nombre compris entre **1** et **100** inclus.

On considère le type **TabEntiers** de la façon suivante :

TYPES

Classe : **STRUCTURE**

tabEtudiants : TABLEAU [1..50] DE CHAINE

nbPresent : ENTIER

FINSTRUCTURE

Ecrire une fonction nommée **NB_OCCURENCES** permettant le calcul du nombre d'occurrences d'un élément **X** donné, dans un tableau **T** passé en paramètre. Compléter la fonction suivante.

FONCTION NB_OCCURENCES (T : T_ENTIERS, X : ENTIER) : ENTIER

VARIABLES

i, nb_occ : ENTIERS

DEBUT

nb_occ ← 0

... A COMPLETER

RETOURNER nb_occ

FIN

Exercice 3 (4 pts)

Ecrire une fonction nommée **PRODUIT_SCALAIRE** permettant le calcul du produit scalaire de deux vecteurs réels **U** et **V** de dimension **N**. Le produit scalaire de deux vecteurs réels $U(U_1, U_2, \dots, U_N)$ et $V(V_1, V_2, \dots, V_N)$ de dimension **N** est défini par :

$$U \cdot V = \sum_{i=1}^N U_i \cdot V_i$$

Compléter la fonction suivante.

FONCTION PRODUIT_SCALAIRE (U : T_ENTIERS, V : T_ENTIERS) : ENTIER

VAR

i, prod_scalaire : ENTIERS

DEBUT

prod_scalaire ← 0

... A COMPLETER

RETOURNER prod_scalaire;

FIN

Exercice 4 (5 pts)

Question 1 :

Ecrire une fonction nommée **CUBE** qui retourne le cube d'un entier **N**. Compléter la fonction suivante.

FONCTION CUBE (x : ENTIER) : ENTIER

VARIABLES

v : ENTIER

DEBUT

... A COMPLETER

FIN

DEVOIR N°1 D'ALGORITHMIQUE

EXERCICE N°1 (4 pts)

Ecrire un algorithme nommé AFFICHE_SAISON qui, à partir du numéro d'un mois saisi par l'utilisateur (1 pour janvier ; 2 pour février ; ... ; 12 pour décembre), permet d'afficher la saison.

Exemple :

Si mois = 6 ou 7 ou 8 Alors le programme affiche : **Été**

Si mois = 3 ou 4 ou 5 Alors le programme affiche : **Printemps**

Si mois = 9 ou 10 ou 11 Alors le programme affiche : **Automne**

Si mois = 12 ou 1 ou 2 Alors le programme affiche : **Hiver**

Compléter l'algorithme suivant :

ALGORITHME AFFICHE_SAISON

VARIABLES

Mois : **ENTIER**

Saison : **CHAINE**

DEBUT

ECRIRE ('Donnez le numéro du mois')

... A COMPLETER

ECRIRE ('Saison = ', Saison)

FIN

Exercice 4 (5 pts)

Considérons les déclarations suivantes :

CONSTANTE

MAX = 100

TYPES

ETUDIANT= STRUCTURE

Nom : CHAINE

Moyenne : ENTIER

Mention : CHAINE

FINSTRUCTURE

T_ETUDIANT=TABLEAU [1..MAX] DE ETUDIANT

Ecrire un Algorithme qui utilise les déclarations précédentes et qui permet de saisir les moyennes de nb étudiants passé en paramètre. À la fin de la saisie, la procédure affichera la moyenne de chaque étudiant suivie de la mention selon le format :

Moyenne de l'étudiant 1:12Mention : **Assez-Bien**

Moyenne de l'étudiant 2:10Mention : **Passable**

...

Etc.

PROCEDURE CONSTRUIRE((E) nb :ENTIER, (S) etudiants : T_ETUDIANT)

VARIABLES locales

i :ENTIER

DEBUT

A COMPLETER

FIN

EXAMEN D'ALGORITHMIQUE**Exercice 1 (5 pts)**

Ecrire un algorithme permettant de saisir des entiers dans un tableau puis calcule le plus grand écart dans ce tableau (l'écart est la valeur absolue de la différence de deux éléments). Compléter l'algorithme suivant.

ALGORITHME PLUS_GRAND_ECART**CONSTANTES**

TAILLE = 100

VARIABLES

t : TABLEAU [0..TAILLE] DE ENTIER

min, max, i : ENTIER

DEBUT**POUR** i **DE** 1 **A** TAILLE **FAIRE**

LIRE(t[i])

FINPOUR... **A COMPLETER****ECRIRE** ("le plus grand écart dans le tableau est : ", max - min)**FIN****Exercice 2 (5 pts)**

Dans la suite, on considère l'algorithme principal suivant :

ALGORITHMIQUE PRINCIPAL**CONSTANTES**

TAILLE_MAX = 100

TYPES

T_ENTIERS = TABLEAU[1..TAILLE_MAX] DE ENTIER

DEBUT

...

FIN

EXERCICE N°4 (4 pts)

Ecrire un algorithme qui prend 50 notes quelconques de l'utilisateur et affiche la moyenne des notes qui sont comprises entre 10 et 20 (bornes incluses).

Compléter l'algorithme suivant :

ALGORITHME Moyenne50Notes**VARIABLES**

NoteSaisi: REEL

SommeDesNotes: REEL

MoyenneDesNotes : REEL

i : ENTIER

DEBUT

SommeDesNotes ← 0

... A COMPLETER

Ecrire ("la moyenne des notes comprises entre 10 et 20 est :", MoyenneDesNotes)

FIN**EXERCICE N°5 (4 pts)**

Ecrivez un algorithme qui lit deux valeurs entières (A et B) du clavier et qui indique si le produit de A et B est négatif ou non sans faire la multiplication.

Compléter l'algorithme suivant :

ALGORITHME ProduitNegatif**VARIABLES**

A, B: ENTIER

DEBUT

Ecrire ("Saisir deux entier A et B :")

... A COMPLETER

FIN

Écrire un Algorithme qui utilise la structure précédente et qui permet de saisir la liste de présence dans une classe quelconque de l'ESATIC.

On posera la question "L'étudiant N° 1 est présent ? (oui/non)". Si la réponse est Oui, On saisit "Oui" dans la cellule N°1 du tableau tabEtudiants. Dans le cas contraire On saisit "Non" dans la cellule N°1 du tableau tabEtudiants. On posera ensuite la question "encore un étudiant à appeler ? (oui/non)". Si la réponse est Oui, on passe à l'étudiant suivant. Dans le cas contraire on sort de l'Algorithme. A la fin de saisi, on pourrait par exemple avoir la configuration suivante pour une classe contenant 9 élèves :

tabEtudiants :

Oui	Non	Oui	Oui	Oui	Non	Non	Oui	Non
-----	-----	-----	-----	-----	-----	-----	-----	-----

nbPresent :

ALGORITHMEStructureTableau

CONSTANTES

TAILLE = 50

TYPES

TABENTIERS= STRUCTURE

tab: TABLEAU[1..TAILLE] DE ENTIERS

nbpresent: ENTIER

FINSTRUCTURE

VARIABLES

Etudiants : TABENTIERS

i : ENTIER /* indice */

reponse : CHAINE /* réponse oui/non */

DEBUT

reponse ← "oui"

A COMPLETER

FIN

Exercice 3 (5 pts)

Considérons le tableau des mentions suivantes :

Moyenne	Mention
Entre 0 et 5 inclus	Faible
Entre 6 et 9 inclus	Insuffisant
Entre 10 et 11 inclus	Passable
Entre 12 et 14 inclus	Assez Bien
Entre 15 et 17 inclus	Bien
Entre 18 et 20 inclus	Très bien

Écrire une fonction nommée MENTION retourne la mention d'un étudiant en fonction de la moyenne passé en paramètre.

FONCTION MENTION (Moyenne:ENTIER):CHAINE

VARIABLES locales

Mention:CHAINE

DEBUT

A COMPLETER

FIN