

Fiche TD-TP n°1 : Notions de base + structures conditionnelles

Rappel des principales consignes pour toute la durée du module :

- En TD on réfléchit en écrivant des algorithmes ;
- Ces algorithmes doivent être correctement indentés, commentés et écrits en utilisant une notation conventionnelle (camelCase pour les variables, UPPER_CASE pour les constantes) ;
- En TP on code en langage C# les algorithmes produits ;
- L'usage d'IA (Chat GPT, Copilot, ...) n'est pas autorisé car serait contre-productif !

La gestion de la mémoire**Exercice 1****Algorithme Ex1**

Entrée : /

Sortie : /

Déclarations

Constantes

E = 5 : entier

F = "bonjour" : chaîne de caractères

Variables

a, b, c, d : entier

e : chaîne de caractères

Début

a = 39

b = a

c = b + 1

d = 0

e = F

a = a + E

b = b + 1

d = d - E

e = e + " à tous !"

Fin

👉 Dérouler l'algorithme Ex1 en utilisant la trace ci-dessous :

E	F	a	b	c	d	e
5	Bonjour					

Exercice 2

Thibaut possède trois seaux :

- Un seau en plastique d'une contenance de 10 litres,
- Un seau en bois d'une contenance de 7 litres,
- Un seau en fer d'une contenance de 9 litres.

10h00 Thibaut vide ses trois seaux en arrosant ses fleurs.

10h05 Thibaut rend visite à Bob (qui possède un puits) qui met 6 litres dans son seau en bois.

10h10 Thibaut transvase le contenu de son seau en bois dans le seau en fer.

10h15 Thibaut revient vers Bob pour remplir à ras bord son seau en plastique.

10h20 Thibaut arrose ses légumes avec la moitié de son seau en plastique.

10h25 Thibaut transvase le contenu de son seau en plastique dans celui en bois.

10h30 Thibaut remplit son seau en fer à ras-bord à partir de son seau en bois.

10h35 Alice demande à Thibaut quel est le contenu de chacun de ses seaux.

a) Remplir la trace suivante :

Heure	Seau en plastique	Seau en bois	Seau en fer
10h00			
10h05			
10h10			
10h15			
10h20			
10h25			
10h30			
10h35			

b) Compléter l'algorithme suivant qui détaille les actions de Thibaut sur ses seaux en utilisant au besoin les constantes et la variable temp :

Algorithme Ex2

Entrée : /

Sortie : /

Déclarations

Constantes

CAP_SP = 10 : entier //Capacité du seau en plastique

...

...

Variables

seauP, seauB, seauF, temp : entier

Début

seauP = 0

...

Fin

La structure séquentielle

Exercice 3

Écrire un algorithme qui calcule et affiche la surface et le périmètre d'un champ rectangulaire d'une longueur de 127m50 et d'une largeur de 58m95.

Exercice 4

Écrire un algorithme qui calcule et affiche l'IMC (indice de masse corporelle) de l'utilisateur après que celui-ci ait saisi sa taille (en mètre) et son poids (en kg). La formule pour une personne adulte est la suivante :

$$imc = \frac{poids}{taille^2}$$

Exercice 5

Modifier l'algorithme de l'exercice 3 pour permettre à l'utilisateur de saisir la longueur et la largeur du champ et d'afficher en plus son prix sachant que le tarif du terrain agricole du département où est utilisée l'application est de 3000 € / ha.

Exercice 6



La CTS (compagnie des transports strasbourgeois) vend un BSC (billet sans contact) chargé avec 10 tickets au prix de 17,30 € TTC. Sachant que le prix du support est de 20 centimes TTC et que les transports bénéficient d'une TVA réduite à 10 %, la CTS souhaite afficher le détail du tarif sous cette forme :

Support : ... € HT
 Tickets : 10 x ... = ... € HT
 Montant TVA (10 %) : ... €
 Total : 17,30 € TTC

- Calculer les valeurs manquantes à l'aide des informations à votre disposition
- Parmi les valeurs affichées dans le détail du tarif, repérer celles qui sont fixes et celles qui sont calculées. Déterminer les constantes et les variables qui seront nécessaires au futur algorithme en complétant le tableau ci-dessous :

Nom conventionnel	Variable	Constante	Type de donnée
PRIX_TOTAL_TTC		x	réel

- Écrire l'algorithme qui affiche le détail du tarif, celui-ci doit pouvoir s'adapter facilement à un changement de tarif du BSC, du prix du support ou de la TVA.

Les structures conditionnelles

Exercice 7

Voici le début d'un algorithme :

Algorithme Ex7

Entrée : /

Sortie : /

Déclarations

Constantes

I1 = 10 : entier

D2 = 67000 : double

F3 = 20.6 : réel

Variables

a, b : entier

c : double

d : réel

enRetard : booléen

Début

//Initialisation des variables

a = 2204

b = 5

c = 1000

d = 200

enRetard = vrai

Lors de cet algorithme différents tests sont effectués, indiquer le résultat de chacune de ces conditions :

Conditions testées	Résultat
Si a == 2204 Alors	
Si c >= F3 Alors	
Si a < 10 ET b <> D2 Alors	
Si a < 10 OU b <> D2 Alors	
Si b == 5 ET c == d Alors	
Si a == 5 OU c == d Alors	
Si NON(F3 > d) Alors	
Si enRetard Alors	
Si I1 < 20 ET NON enRetard Alors	
Si c == 1000 OU (F3 == 68000 ET enRetard) Alors	
Si (c == 1000 ET F3 == 68000) OU enRetard Alors	
Si c == 1000 ET F3 == 68000 OU NON enRetard Alors	
Si c <> 1000 ET F3 <> 68000 OU NON enRetard Alors	

Remarque : Si enRetard Alors //idem à Si enRetard == vrai Alors
 Si NON enRetard Alors //idem à Si enRetard == faux Alors

Exercice 8

Écrire un algorithme qui demande à l'utilisateur de saisir son âge. L'algorithme doit contrôler si la valeur saisie est comprise entre 0 et 122 ans. 2 cas de figure :

- La saisie est correcte, l'algorithme affiche si la personne est mineure ou majeure,
- La saisie n'est pas correcte, l'algorithme affiche un message explicite à destination de l'utilisateur.

Exercice 9

Selon les classifications de l'OMS, l'IMC se répartit comme suit :

- Moins de 16,5 : insuffisance pondérale sévère (niveau -2)
- de 16,5 à 18,5 : insuffisance pondérale (niveau -1)
- 18,5 à 25 : Corpulence normale (niveau 0)
- 25 à 30 : Surpoids (niveau 1)
- 30 à 35 : Obésité de grade 1 dite modérée (niveau 2)
- 35 à 40 : Obésité de grade 2 dite sévère (niveau 3)
- Plus de 40 : Obésité de grade 3 dite morbide (niveau 4)

- a) Etablir l'arbre qui schématise les tests à effectuer pour déterminer le niveau de l'IMC.
- b) Compléter l'algorithme de l'exercice 4 afin de vérifier que la taille saisie est bien comprise entre 1m50 et 2m10. Si la taille est valide l'algorithme doit afficher, en plus de l'IMC, le niveau et le libellé associés.

Exercice 10

Si on considère un système 32 bits qui utilise, pour stocker une date, le format UNIX timestamp, la date la plus ancienne représentable est le 13/12/1901 et la date la plus lointaine est le 19/01/2038 (cf Bug de l'an 2038). Écrire un algorithme qui demande à l'utilisateur de saisir 3 entiers correspondants au jour, mois et année d'une date et qui indique si cette date est valide. Pour être valide la date doit remplir obligatoirement les critères suivants :

- Le mois doit être compris entre 1 et 12 ;
- Le jour doit être un jour autorisé pour le mois saisi (le 30 février, le 31 avril ou le 45 juin seront considérés invalides par exemple) ceci en tenant compte des années bissextiles ;
- La date doit être comprise entre le 13/12/1901 et le 19/01/2038

L'algorithme devra afficher la date (au format **dd/mm/yyyy**) et indiquer si elle est valide ou non, dans ce cas il faudra préciser les causes de non validité.



Une année est bissextile si elle divisible par 4 mais pas par 100 ou divisible par 400 (par exemple 2000 était bissextile, 1900 ne l'était pas).