



AUTORISATION N° 64/21 MINESEC/SG/DESG/SDSGEPESG/SSGEPESG DU 26/07/2021

Fiche d'intégration : algorithmique & programmation en C

CLASSE : TERMINALE C-D

Exercice 1 : Quelques rappels de la seconde

1. Définir les termes suivants : **Algorithme, algorithmique, programme, variable, constante, incrémentation, décrémentation, organigramme, affectation, instruction, compilateur, IDE, programmation**
2. Présenter les caractéristiques d'une variable.
3. Enumérer les différentes structures de contrôle utilisées en algorithmique.
4. Dessiner les symboles utilisés en algorithme pour désigner respectivement :
 - ❖ Un début ou fin
 - ❖ Une entrée ou sortie
 - ❖ Un traitement
 - ❖ Une condition entraînant plusieurs choix.
5. Donner le symbole utilisé en algorithmique pour représenter l'opérateur d'affectation.
6. Enumérer trois instructions simples utilisées en algorithmique.

Exercice 2 : Les algorithmes simples pour commencer.

On considère l'algorithme ci-dessous :

Algorithme *Exemple*

Var a, b, s : entier ;

Début

Ecrire ("entrer un nombre :") ;

Lire(a) ;

Ecrire ("entrer un autre nombre :") ;

Lire(b) ;

$s \leftarrow (a+b)*2$;

Ecrire (" le résultat du calcul est :", s) ;

Fin

1. Quel est le nom de cet algorithme.
2. Donner la liste des variables déclarées dans cet algorithme.
3. Identifier dans cet algorithme :
 - Les opérateurs arithmétiques

- Les instructions de lecture et d'écriture
 - Le nombre d'instructions utilisées dans cet algorithme.
4. Quel est la structure utilisée dans cet algorithme.
 5. Dessiner l'organigramme de cet algorithme.

Exercice 3 : restons *toujours dans les algorithmes simples*.

Vous souhaitez aider votre papa à calculer de façon automatique la surface de son champ qui a la forme d'un trapèze en écrivant un algorithme qui va résoudre ce problème.

1. Proposer un nom correct de l'algorithme que vous allez écrire.
2. En utilisant la formule mathématique de calcul de la surface de trapèze, identifier les variables qui seront utilisées dans votre algorithme puis donner leur type.
3. Écrire alors cet algorithme.

Exercice 4 : les *algorithmes utilisant la structure alternative*.

1. Ecrire un algorithme permettant de calculer et afficher la valeur absolue d'un nombre.
2. Ecrire un algorithme qui prend en entrée la moyenne d'un élève de classe de 1^{ère} D et renvoi la décision **Admis** ou **Refusé** selon les cas suivants : **Admis** si la moyenne est supérieure ou égale à 10 **Refusé** dans le cas contraire.
3. On souhaite écrire un algorithme permettant de déterminer l'état de l'eau en prenant pour paramètre sa température tel que donnée ci-dessous :
Glacé : température inférieure à 0°C
Liquide : température supérieure à 0 et inférieure à 100°C
Vapeur : température supérieure à 100 °C Ecrire l'algorithme permettant de résoudre ce problème.

Exercice 5 : les *structures itératives*.

On souhaite écrire un algorithme qui prend entrée les notes en informatiques de 50 élèves d'une classe de 1^{ère} C puis calcule la moyenne de ces notes, tout en retournant le nombre des élèves qui ont une moyenne supérieure ou égale à 10.

1. Donner la liste des variables à utiliser dans cet algorithme en précisant leur type.
2. Identifier les structures à utiliser dans à cet algorithme.
3. Ecrire alors cet algorithme.

Exercice 6 : une autre façon de faire la multiplication

```
Algorithme Exo ;
Var a,b,p,i : entier ;
Début
    Ecrire ("donner un nombre");
    Lire(a) ;
    Ecrire ("donner un autre ");    Lire(b)
;
    Si (a=0 ou b=0) Alors p←0;
        Sinon p←0;
            Pour i allant de 1 à b Faire p←p+a ;
    FinPour
Finsi
Ecrire(p) ;
Fin
```

On considère l'algorithme ci-contre :

1. Donner la liste des variables et leur type.
1. Compter dans cet algorithme :
 - Le nombre des instructions
 - Le nombre d'affectation
3. Identifier les structures de contrôle utilisé dans cet algorithme.
4. Donner le contenu de la variable p en sortie pour chacune des cas suivants :
 - Cas 1 : a=0 et b=0
 - Cas 2 : a=4 et b=4
 - Cas 3 : a=5 et b=6.
 - Cas 4 : a=2 et b=8
5. Déduire de la question précédente ce que fait cet algorithme.
6. Ecrire une autre version d'algorithme plus simple réalisant la même tâche que celui-ci.

Exercice 6 : Structures des données

On a utilisé le tableau pour stocker les notes en mathématiques des élèves d'une classe de 1^{ère} E tel que donné ci-dessous :

12	03	11	09	18	12	08	05	04	15
----	----	----	----	----	----	----	----	----	----

1. Définir tableau, structure des données.
2. Enumérer 03 autres structures des données que vous connaissiez.
3. Citer les éléments caractéristiques d'un tableau.
4. Donner la dimension et la taille du tableau ci-dessous.
5. Ecrire un algorithme qui demande à l'utilisateur d'entrer un par un les données de ce tableau.
6. Ecrire un algorithme qui permet de faire la somme des éléments de ce tableau.
7. Ecrire un algorithme qui renvoi la note maximale et minimale parmi celles stockées dans le tableau ci-dessous.

Exercice 7 : Les sous programmes

1. Définir les termes suivants : sous programmes, fonction, procédure.
2. Donner l'intérêt de l'utilisation d'un sous-programme en algorithmique.
3. Présenter la différence entre les paramètres formels et paramètres effectifs.
4. Ecrire un sous-programme qui étudie la parité d'un nombre.
5. Ecrire une fonction récursive (puis itérative) qui calcule le terme n de la suite de Fibonacci définie par : $U_0=U_1=1$; $U_n=U_{n-1}+U_{n-2}$

6. Ecrire une fonction qui calcule x^n .

Exercice 8 : un peu de théorie sur le langage C.

1. Définir les mots et expressions suivantes : **programme, compilateur, langage de programmation.**
2. Enumérer 03 exemples de compilateurs C.
3. Donner les rôles des bibliothèques C suivants : `stdio.h`, `stdlib.h`, `math.h` et `conio.h`.
4. Donner les rôles des fonctions C suivantes : `scanf`, `printf`, `get`.
5. Que signifie IDE. Donner deux exemples.
6. Donner les mots clés utilisés en langage C pour designer :
 - a. Un entier
 - b. Un réel
 - c. Un caractère
 - d. Un booléen
7. Recopie et complète le tableau ci-dessous

Operateurs	Symboles
	=
Reste de la division (modulo)	
	<=
Différence	
égalité	

8. Donner le rôle des symboles `%` et `&` utilisés dans les instructions d'entrée/sortie en langage C
9. Présenter la différence entre la variable globale et la variable locale.

Exercice 9 : des programmes en langage C.

Soient les programmes C suivant :

Programme 1	Programme 2
<pre> #include <Stdio.h> int main (void) { int x, abs; abs=((x>=0)? x : -x) ; printf(" abs= %d",abs); } Return 0; } </pre>	<pre> #include <Stdio.h> int main (void) { int a, b ; b=((a=3),(a+2)) ; printf(" b= %d",b); } Return 0; } </pre>

1. Donner le rôle des opérateurs `?` et **virgule (,)**
2. Qu'affiche le programme 2 après son exécution.
3. Qu'affiche le programme 1 si on l'exécute avec la valeur `x= -10` ?

- Réécrire le programme 1 en utilisant la structure alternative (if...else).

Exercice 10 : je deviens déjà programmeur en langage C

- Ecrire un programme C permettant de créer le tableau ci-dessous et de rechercher dans ce tableau une valeur fournie par l'utilisateur.

12	3	4	0	15	25
----	---	---	---	----	----

Si la valeur est trouvée, alors le message suivant est affiché : « Nombre trouvé ! ». Dans le cas contraire affiché « Nombre non trouvé ! »

- Ecrire un programme C qui calcule et affiche le PGCD de deux nombres fournis par l'utilisateur.
- Ecrire un programme C qui affiche la table de multiplication d'un nombre fourni par l'utilisateur.

Exercice 11 : calcul de la moyenne

Soit le programme C suivant :

Pour pouvoir calculer facilement sa moyenne, votre petit frère a écrit le programme C ci-contre.

```

1
2  #include <stdio.h>
3  #include<stdlib.h>
4
5  int main(){
6
7      int Note[5]={10,20,2,13,15};
8      int coef[5]={1,2,2,3,4};
9      int i=0,somme=0,tCoef=0;
10     double moyenne;
11
12     while(i<=4){
13         somme = somme+Note[i]*coef[i];
14         i++;
15     }
16
17     for(i=0;i<=4;i++){
18         tCoef = tCoef +coef[i];
19     }
20
21     moyenne = somme/tCoef;
22
23     printf(" la moyenne est %f \n",moyenne);
24
25     return 0;
26 }

```

- Combien de tableau a-t-on déclaré dans ce programme ?
- Identifier toutes les variables déclarées dans ce programme puis donner leur type.
- Donner les boucles utilisées dans ce programme.
- Donner le nombre d'initialisation contenu dans ce programme.
- Expliquer ce que fait les parties de ce programme :
 - De la ligne 12 à la ligne 15.
 - De la ligne 17 à la ligne 19.
- Exécuter ce programme puis donner la valeur finale de la variable **moyenne**.

BON TRAVAIL...