

---

# TD d'informatique

Florent Bruguier - Joël Durand - Eric Nativel - 5 septembre 2017

---

---

# 1 - Exercices sur l'affectation interne

Pour chacune des séquences données ci-dessous, indiquez les valeurs des variables après chaque affectation réalisée.

Exemple :

DEBUT

$A \leftarrow 1$     *A reçoit la valeur 1*

$B \leftarrow 3$     *B reçoit la valeur 3*

$A \leftarrow B$     *A reçoit la valeur 3*

FIN

A	B
1	?
1	3
3	3

---

DEBUT

$A \leftarrow 1$

$B \leftarrow 3$

$B \leftarrow A$

$A \leftarrow B$

FIN

---

DEBUT

$A \leftarrow 5$

$B \leftarrow 2$

$C \leftarrow A+B$

$A \leftarrow 2$

$C \leftarrow A / B$

FIN

---

DEBUT

$A \leftarrow 5$

$B \leftarrow A+4$

$A \leftarrow A+1$

$B \leftarrow A-4$

FIN

---

---

## 2 - Primitives d'Entrées / Sorties : LIRE et ECRIRE

Décrire l'effet produit par chaque instruction des algorithmes suivants. Indiquer les affichages qu'ils génèrent. Proposez la traduction en langage C.

*Exemple :*

DEBUT

A ← 1

*A reçoit la valeur 1*

ECRIRE A

*Le chiffre « 1 » s'affiche à l'écran*

FIN

---

DEBUT

A ← 1

ECRIRE "valeur ?"

LIRE A

ECRIRE A

FIN

---

DEBUT

A ← 1

ECRIRE "valeur ?"

LIRE A

ECRIRE "A"

FIN

---

DEBUT

A ← 1

LIRE A

ECRIRE "A ← 2"

ECRIRE A

FIN

---

## Exercice 2B - Analyse d'algorithmes

Reproduire les messages affichés par les algorithmes suivants lors de leur exécution.

-----  
Variables a, b, c : nombres entiers

Debut

a ← 1

b ← 2

c ← 3

---

```
a ← b
b ← c
c ← a
```

```
Ecrire "a=", a
Ecrire "b=", b
Ecrire "c=", c
```

Fin

---

Variables a, b, c : nombres entiers

Debut

```
a ← 4
b ← 5
c ← 6
```

```
a ← a+b
b ← a-b
a ← a-b
a ← a+c
c ← a-c
a ← a-c
```

```
Ecrire "a=", a
Ecrire "b=", b
Ecrire "c=", c
```

Fin

---

---

## 3 - Exercices sur les expressions logiques

1. Soit « n » une variable de type nombre entier. Déterminer l'écriture d'une expression logique qui soit

- VRAIE si « n » contient une valeur divisible par 4
  - FAUSSE sinon
- 

2. Soit « a » et « b » deux nombres tels que  $a < b$ . Déterminer l'écriture algorithmique équivalente aux expressions logiques suivantes :

x appartient à [a,b]

x n'appartient pas à [a,b]

---

3. Soit « N1 » et « N2 » deux nombres. Déterminer l'écriture algorithmique de l'expression logique qui est vraie si l'une des deux valeurs est supérieure à 10

---

4. Soit x, y, z, t quatre variables numériques d'un environnement donné. Chacune des questions suivantes caractérise une situation de cet environnement. Exprimez les expressions booléennes qui correspondent à ces situations :

d1- Les valeurs de x et de y sont toutes les deux supérieures à 3

d2- Les variables x,y et z contiennent toutes des valeurs identiques

d3- Les valeurs de x,y et z sont identiques mais différentes de celle de t

d4- La valeur de x est strictement comprise entre les valeurs de y et t

d5- La valeur de x est strictement comprise entre les valeurs de y et t et la valeur de y est inférieure à celle de t

d6- Parmi les valeurs de x,y et z, deux valeurs au moins sont identiques

d7- Parmi les valeurs de x,y et z, deux valeurs et seulement deux sont identiques

---

## 4 - Exercice avec les types.

1. Évaluer les expressions suivantes en supposant à chaque fois que l'on a déclaré les variables suivantes :

```
int A=20, B=5, C=-10, D=2, X=12, Y=15 ;
```

Donner la valeur de la variable A.

On suppose que vrai correspond à la valeur 1 et 0 correspond évidemment à faux.

- $A = (5 * X) + 2 * ((3 * B) + 4)$
- $A = (5 * (X + 2) * 3) * (B + 4)$
- $A = (A == B)$
- $A += (X + 5)$
- $A = (A != (C * -D))$
- $A *= C + (X - D)$
- $A \% = D ++$
- $A \% = ++D$
- $A = (X ++ ) * (A + C)$
- $A = !(X - D + C) \parallel (D - 1)$

2. On définit cette fois-ci les variables suivantes :

```
int compteur=10, nbreA=100, nbreB=51 ;  
float prix=20.30, ptu ;  
char car ='A' ;
```

Evaluer les expressions suivantes :

- $\text{Prix} += \text{compteur} * 0.196 ;$
- $\text{Compteur} = \text{prix} / \text{nbreA} ;$
- $\text{NbreA} = \text{prix} / \text{NbreB} ;$
- $\text{NbreB} = \text{ptu} ++ ;$
- $\text{Compteur} = \text{car} ++ ;$
- $\text{car} = \text{car} + \text{NbreB} ;$
- $\text{ptu} = \text{prix} + \text{car} / \text{NbreA} ;$
- $\text{compteur} = \text{prix} * \text{NbreB} * \text{NbreB} * \text{NbreB} ;$
- $\text{car} = 20 * \text{prix} ;$
- $\text{car} = 'A' + 'C' ;$

---

## 5 - Exercices d'analyse et d'algorithmique

Pour chacun des problèmes proposés ci-dessous :

Phase 1

- 1) Identifiez les données et les résultats attendus. Préciser le type de chaque information identifiée et proposez un identificateur de variable pour chacune d'elle.
- 2) Définissez le traitement permettant de résoudre le problème. Si nécessaire, complétez l'environnement avec des variables auxiliaires, nommées et typées.

Phase 2

- 3) Proposez un algorithme de résolution du problème

1 . A partir d'une quantité de produit achetée à un prix hors taxe on veut établir la facture d'un client dont on connaît le nom. La facture fera apparaître le nom, le montant hors taxe, et le montant toute taxe sachant qu'on applique un taux de taxe de 10%.

2. La suie peut être utilisée comme engrais : on peut l'employer à la dose de 1/3 de mètre cube par are.

On cherche quelle serait la dépense à faire pour enrichir un nombre X d'ares. La suie coûtant 4,50 euros l'hectolitre, les frais de transports étant de Y euros les 10 kilos, sachant qu'un mètre cube de suie pèse 1205 kg.

3. Une boulangerie est ouverte tous les jours de 6h à 13h et de 16h à 19h, sauf le dimanche après-midi et le lundi toute la journée. On veut savoir pour un jour et une heure donnés, si cette boulangerie est ouverte ou non. (On ne prendra pas en compte le nombre des minutes pour traiter le problème)

On pourra proposer deux solutions : une qui utilise la structure alternative, l'autre qui est uniquement basée sur l'affectation.

4. On veut déterminer la saison correspondant à une date donnée sous la forme J/M/A. On considère que les changements de saisons se produisent les jours suivants :

21 mars:	début du printemps
21 juin:	début de l'été
21 septembre:	début de l'automne
21 décembre :	début de l'hiver

---

## Annexe : étapes de la mise en oeuvre d'un algorithme

### Phase 0 : Énoncé

On veut calculer la surface d'un terrain rectangulaire ...

### Phase 1 : Etape de réflexion

A partir de l'énoncé on doit définir les flux entrants (les données du problème), les flux sortants (les résultats du problème), et le moyen de passer des uns aux autres.

Pour cet exemple, on détermine aisément le résultat car il fait partie de l'énoncé (la surface d'un rectangle), ce qui n'est pas le cas des données.

On va donc mettre en place un calcul permettant de déterminer le résultat (de façon simple) ce qui fera apparaître par décomposition fonctionnelle les données indispensables.

Résultat:      SURF représente la surface du terrain en m<sup>2</sup> (nombre réel)

Données :      LONG représente la longueur du terrain en m (nombre réel)  
                  LARG représente la largeur du terrain en m (nombre réel)

Traitement :   SURF ← LONG x LARG

### Phase 2 : L'algorithme et son environnement

#### Lexique

LONG (nombre réel) La longueur du rectangle, exprimée en mètres

LARG (nombre réel) La largeur du rectangle, exprimée en mètres

SURF (nombre réel) La surface du rectangle, exprimée en mètres carrés

#### Début

ECRIRE « Donner la longueur puis la largeur du terrain : »)

Lire (LONG, LARG)

-----  
SURF ← LONG \* LARG

-----  
Ecrire (« La surface du terrain est de », SURF, « m<sup>2</sup> »)

#### Fin

### Phase 3 : Codage

Ici, on traduit l'algorithme en langage C et on génère le programme exécutable. (

```
// -----  
  
// Calcul de la surface d'un terrain rectangulaire  
// -----  
  
#include <stdio.h>  
#include <stdlib.h>  
  
void main(void)  
{  
    float long, larg, surf;
```



---

```
printf("donner la longueur du terrain:");
scanf("%f",&long);
printf ("donner la largeur du terrain:");
scanf("%f",&larg);

surf = long * larg ;

printf("\r La surface du terrain est de %f m2\n", surf) ;

}
```

#### Phase 4 : Jeux d'essais (tests)

On note les résultats générés par l'exécution du programme pour plusieurs cas.

1ere saisie	2eme saisie	Valeur affichée
10	20	
0	10	
10	0	
10.1	20	
10,1	20	
Toto	20	

---

## 6 - Exercices sur les entrées-sorties conversationnelles

1. Écrire un programme C qui calcule la surface et le volume d'une sphère à partir du rayon qui sera lu au clavier. Les sorties devront se faire avec un affichage sur 10 caractères, dont 2 après la virgule.
2. Écrire un programme C qui permet de déterminer le minimum de quatre nombres.
3. Ecrire un programme en C qui calcule de prix de revient d'un carton de lessive contenant n paquets de lessive de 2 kg chacun sachant que le prix est calculé de la sorte : un prix de base 500g de lessive vaut 50 cents auquel se rajoute une taxe est fonction du taux de change à la bourse soit 5,5% par kg de lessive.

## 7 - Exercices avec les structures conditionnelles

1. Réécrire l'algorithme suivant en explicitant correctement l'indentation des blocs

### Variables

a, b, c, d : nombres entiers

Debut

a ← 16

b ← 4

c ← 3

SI a MOD b = 0 ALORS

Début

d ← 0

SI c≠3 ALORS

Début

d ← 1

Fin

SINON

Début

SI c≠4 ALORS

Début

d ← 2

Fin

SINON

Début

d ← 3

---

Fin  
Fin  
SI  $c \neq 5$  ALORS  
Début  
 $d \leftarrow 4$   
Fin  
Fin  
SINON  
Début  
 $d \leftarrow 5$   
Fin  
Fin

Donner la valeur finale de la variable  $d$  à l'issue de l'exécution de l'algorithme.

2. Voici deux algorithmes qui affichent un message en fonction de la température qu'il fait.

Algorithme 1

Début  
Ecrire « Quelle température fait-il ? »  
Lire  $temper$   
si  $temper < 18$  alors  
     $sensation \leftarrow$  « plutôt frais »  
finsi  
si  $temper > 18$  alors  
     $sensation \leftarrow$  « plutôt bon »  
finsi  
Ecrire « il fait »,  $sensation$   
Fin

Algorithme 2

Début  
Ecrire « quelle température fait-il? »  
Lire  $temper$   
si  $temper < 18$  alors  
     $sensation \leftarrow$  « plutôt frais »  
sinon  
     $sensation \leftarrow$  « plutôt bon »  
finsi  
Ecrire « il fait »,  $sensation$

Fin

Pour ces 2 algorithmes remplir l'affichage produit en fonction des valeurs saisies

Valeur saisie	Affichage produit	
	Via algorithme 1	Via algorithme 2
15		
18		
22		

Quelle modification peut on apporter à l'algorithme 1 pour qu'il génère les même résultats que l'algorithme 2?

L'algorithme 3 est censé afficher :

il fait plutôt frais

Lorsque la température est inférieure à 18°

il fait plutôt bon

lorsque la température est comprise entre 18° et 25°

il fait plutôt chaud

lorsque la température dépasse les 25°

### Algorithme 3

Début

Ecrire « quelle température fait-il ? »

Lire temper

Si temper<18 alors

sensation←« plutôt frais »

Finsi

Si temper>25 alors

sensation←« plutôt chaud »

sinon

sensation←« plutôt bon »

Finsi

Ecrire « il fait », sensation

Fin

a) Est-ce bien le cas?

b) Quelle(s) modification(s) permettrai(en)t d'obtenir le comportement attendu? plusieurs solutions sont possibles ....

3. On veut déterminer l'impédance équivalente d'un dipôle formé par 2 résistances pouvant être montées en série ou en parallèle.

4. On veut convertir une valeur de température de °C en °F et vice-versa<sup>1</sup>.

L'utilisateur doit pouvoir choisir le sens de la conversion à réaliser.

---

<sup>1</sup>  $T(^{\circ}\text{C}) = 9/5 * (T(^{\circ}\text{F}) - 32)$

---

## 5. Algorithme de calendrier perpétuel (source <http://www.recreomath.qc.ca/>)

Grâce aux trois tableaux ci-dessous et à l'algorithme donné à la suite, on peut déterminer le jour de la semaine qui correspond à une date donnée ...

Tableau 1. Code du mois

Mois	janv.	fév.	mars	avril	mai	juin	juil.	août	sept.	oct.	nov.	déc
Code	01*	04*	04	00	02	05	00	03	06	01	04	06

\* Si l'année est bissextile, janvier correspond à 00 et février à 03.

Tableau 2. Code du siècle

Siècle	1800	1900	2000	2100	2200	2300
Code	02	00	06	04	02	00

Tableau 3. Code du jour de la semaine

Jour	dimanche	lundi	mardi	mercredi	jeudi	vendredi	samedi
Code	01	02	03	04	05	06	00

Voici l'algorithme à appliquer :

1. Additionner
  - a) le quantième du mois
  - b) le code du mois (tableau 1)
  - c) le code du siècle (tableau 2)
  - d) les deux derniers chiffres de l'année
  - e) ce dernier nombre divisé par 4 en ignorant le reste

On obtient un total

2. Faire la division Euclidienne par 7 du total obtenu et conserver seulement son reste.

3. Utiliser le Tableau N° 3 pour déduire le jour de la semaine à partir du résultat de l'opération précédente

6. Une boulangerie est ouverte tous les jours de 6h à 13h et de 16h à 19h, sauf le dimanche après-midi et le lundi toute la journée. Donner un algorithme qui permet de savoir pour un jour et une heure donnés, si cette boulangerie est ouverte ou non. (On ne prendra pas en compte le nombre des minutes pour traiter le problème)

7. On veut déterminer la saison correspondante à une date donnée.

On rappelle que les saisons sont définies entre les dates suivantes :

---

Du 21/3 au 20/6 -> printemps  
 21/6 au 21/9 -> été  
 22/9 au 20/12 -> automne  
 21/12 au 31/12  
 et du 1/1 au 20/3 -> hiver

## 8. Analyse du « parenthésage » d'une expression

### Principe :

Une expression « parenthésée » est correctement constituée si :

- elle comporte le même nombre de parenthèses ouvrantes que de parenthèses fermantes
- en tout point de l'expression parcourue de gauche à droite, le nombre de parenthèses fermantes rencontrées n'est jamais supérieur au nombre de parenthèses ouvrantes rencontrées.

### Mise en oeuvre du principe :

L'expression parenthésée est donnée sous la forme d'une chaîne de caractères à zéro terminal. On parcourt cette expression de la gauche vers la droite jusqu'à trouver le zéro terminal et on s'appuie sur un compteur préalablement initialisé à 0.

Pendant le parcours, on incrémente le compteur d'une unité chaque fois qu'une parenthèse ouvrante est rencontrée et l'on décrémente ce même compteur d'une unité chaque fois qu'une parenthèse fermante est rencontrée.

La cohérence de l'expression sera établie :

- si la valeur finale du compteur est nulle (condition 1)
- ceci sous réserve que le compteur n'ait jamais pris de valeur négative pendant le parcours (condition 2)

### Environnement

Identificateur	Type / Rôle
Expression	Donnée, chaîne de caractères AZT contenant l'expression à analyser
Index	Auxiliaire, entier, index de parcours de la chaîne
Compteur	Auxiliaire de parcours de la chaîne, entier, augmente de 1 lorsqu'une « ( » est trouvée diminue de 1 lorsqu'une « ) » est trouvée
Cohérence	résultat booléen, reflète la cohérence de l'expression « Expression »

### Algorithme

#### Début

Ecrire « Entrer une expression parenthésée : »

Lire Expression

Positionner Index sur le premier caractère de Expression

Compteur ← 0

Cohérence ← VRAI

Tant que le caractère situé en position Index de Expression n'est pas la zéro terminal, faire

```

Si caractère situé en position Index de Expression est « ( » alors
    ajouter 1 à Compteur
Finsi
Si caractère situé en position Index de Expression est « ) » alors
    retrancher 1 à Compteur
Finsi
si Compteur<0 alors
    Cohérence ←—FAUX
Finsi
positionner Index sur le caractère suivant de Expression
Fin tantque
Si Compteur différent de 0 alors
    Cohérence←— FAUX
Finsi
Ecrire Cohérence
Fin

```

### Travail demandé :

1)

a) Implémenter cet algorithme en langage C.

b) Comment se comporte le programme lorsque l'incohérence d'une longue expression est détectée au tout début de cette expression ?

Exemple :  $a/(b-2)-4*((1+4+5+8+7+9+2+4+8)/((2+6)/(5*3)))$

Proposer une modification de l'algorithme permettant d'éviter de poursuivre inutilement le parcours de la chaîne à partir du moment où l'incohérence est détectée.

Modifier le programme en conséquence et vérifier son fonctionnement.

2) Afin de mieux mettre en évidence la structure de l'expression parenthésée, on souhaite transformer son écriture initiale, en une écriture qui utilise une alternance de symboles :

() (parenthèses)

{ } (accolades)

[ ] (crochets)

Ainsi :

L'expression  $(a-b) / (c*(d+((e*f)/(g+h))))$

devient  $(a-b) / (c* \{d+[(e*f)/(g+h)]\})$

Proposer un principe qui permet d'obtenir une telle transformation.

Ecrire et mettre au point le programme qui permettra :

- d'acquérir une expression parenthésée
- de déterminer si elle est ou non cohérente
- et qui dans le cas où elle est cohérente, la transformera comme ci-dessus.

### 9. Péage autoroutier

A une sortie d'autoroute, les usagers présentent une carte sur laquelle sont indiqués la catégorie du véhicule et le kilométrage de la gare d'entrée sur l'autoroute.

Il y a trois catégories de véhicules: 2 roues, véhicules légers et poids lourds. A chacune de ces catégories correspond un prix au kilomètre:

catégorie 1 (2 roues): 25 cts/km

---

catégorie 2 (véhicules légers):	40 cts/km
catégorie 3 (poids lourds):	52 cts/km

Chaque gare de sortie (ou d'entrée) est parfaitement déterminée par son kilométrage par rapport à l'origine de l'autoroute (prise dans le sens de circulation des véhicules pour la voie concernée).

Proposer un algorithme qui permet d'indiquer à un usager le montant à payer.

10. On se propose de trouver un algorithme qui permet de savoir si une cible rectangulaire a été touchée. Cette cible a pour dimension  $(a \times b)$ . On testera alors le fait d'avoir un point de coordonnées  $(x,y)$  (fléchette) dans ce carré ou non. Lorsque la cible est touchée on proclamera le vainqueur (« gagné ») et on affichera « perdu » dans le cas contraire ainsi que la distance entre le point choisi et le carré.

Variante du cas précédent, on a affaire à des carrés de dimensions de plus en plus grandes : chacun des carrés a des dimensions doubles du précédent.

Si on touche le plus petit carré (dimensions :  $a \times a$ ), on obtient 100 points, si on touche le deuxième, 75 points, le troisième 50 points et ainsi de suite. Lorsque l'on touche un carré qui vaut 0 point, on proclamera le vaincu (« perdu »).

11. Calcul de la durée d'un vol

Ecrire les différentes versions d'un algorithme qui permet de déterminer la durée d'un vol connaissant l'heure de départ et l'heure d'arrivée. (*réponses à porter en page 8*)

Première version:

On considère que le départ et l'arrivée ont lieu même jour et que le temps de vol doit être exprimé en minutes.

exemple: Départ à 6h50mn, arrivée à 9h03 donnerait un temps de vol de 133 mn.

Deuxième version:

On considère que le départ et l'arrivée ont lieu même jour et que le temps de vol doit être exprimé en heures- minutes.

exemple: Départ à 6h50mn, arrivée à 9h03 donnerait un temps de vol de 2h 13mn.

Troisième version:

On considère que la durée de vol est inférieure à 24 heures mais que l'arrivée peut avoir lieu le lendemain. Le temps de vol doit être exprimé en heures- minutes.

exemple: Départ à 23h50mn, arrivée à 9h03 donnerait un temps de vol de 9h 13mn



---

## 8 - Exercices sur les structures itératives

A - Etude d'un algorithme utilisant la boucle « tant que »  
soit l'algorithme suivant :

```
constante  STOP←-1
variable   val, totalValeurs
Début
1   totalValeurs←0
2   Ecrire « Donner une valeur », STOP, « pour finir »
3   Lire val
4   tant que val est différent de STOP faire
5       totalValeurs←totalValeurs+val
6       Ecrire « Donner une autre valeur », STOP, « pour finir »
7       Lire val
      Fin tant que
9   Ecrire « La somme des valeurs saisies est : », totalValeurs
Fin
```

Lors de son exécution on suppose que l'utilisateur introduit successivement les valeurs suivantes :

12, 5, 6, -1

a) Dans ces conditions, terminez de compléter la table ci-dessous en indiquant dans la première colonne la séquence des N° de lignes traitée par le processeur, et en précisant pour chaque ligne, les valeurs prises par les variables à l'issue de l'exécution des instructions rencontrées.

N° Ligne	val	totalValeurs
début	?	?
1	?	0
2	?	0
3	12	0

---


b) Décrivez la fonction de cet algorithme.

c) Proposez un équivalent à cet algorithme mais en le construisant avec une « boucle répéter ... tant que ... »

A-bis - Décrire la fonction des algorithmes suivants :

a)

```
    Ecrire « Donner une valeur positive paire : »
    Lire valeur
    Tant que valeur < 0 ou valeur % 2 est différent de 0 faire
        Ecrire « Donner une valeur positive et paire »
        Lire valeur
    Fin tant que
```

b)

```
somme ← 0
répéter
    Lire val
    somme ← somme + val
tant que somme <= 500
```

c)

```
    Lire val
    tant que val est différent de 0
        Ecrire val * val
        Lire val
    Fin tant que
```

B - utilisation des boucles pour le contrôle des entrées

1. Reprendre le programme de calcul de saison et le modifier de façon à faire en sorte que l'utilisateur soit forcé à donner une valeur de jour comprise entre 1 et 31, ainsi qu'une valeur de mois comprise entre 1 et 12.

2. Proposer une solution pour contrôler la validité du couple « jour, mois » (on considèrera que février compte toujours 28 j.) En déduire un programme qui force son utilisateur à fournir une date 'valide'.

---

## C - Traitement de listes de données

### 1. Liste de longueur fixée dans l'énoncé du problème

En utilisant une seule variable pour la saisie, proposer un programme qui permet à son utilisateur de calculer la somme d'une liste de 10 valeurs entières.

### 2. Liste de longueur fixée par l'utilisateur

Sur le même principe, proposer un programme qui permet à son utilisateur de calculer la somme d'une liste de 'N' valeurs entières, N étant une valeur fournie par l'utilisateur en début de programme.

### 3. Liste dont la fin est détectée par une valeur particulière (sentinelle)

Proposer un programme qui permet à son utilisateur de calculer la somme d'une liste de valeurs strictement positives fournies par l'utilisateur, la fin de la liste étant détectée lors de la saisie d'une valeur négative.

## D - Jeu de dés

Deux Joueurs lancent deux dés tour à tour. Les points sont comptés de la façon suivante :

- un double six rapporte 36 points
- un double autre rapporte 6 points sauf le double 1 qui fait perdre 12 points
- deux dés « pairs » rapporte 3 points
- toutes les autres combinaisons ne rapportent pas de point.

Proposer un algorithme de calcul du nombre de points associé à un lancé donné puis pour que les lancés s'arrêtent lorsque le premier joueur atteint 44 points : il gagne la partie.

Indications :

Le lancé de dés sera simulé par tirage aléatoire de deux valeurs comprises entre 1 et 6 (voir rand, srand ..)

En début de partie, chaque joueur sera invité à donner son nom de façon à ce que la machine puisse inviter nominativement chaque joueur à lancer les dés.

En fin de partie, la machine proposera aux joueurs d'en faire une autre. Dans l'affirmative, c'est le joueur qui jouait en second à la partie précédente qui commencera à jouer en premier pour la nouvelle partie.

---

## E - Table de multiplication

Proposer l'algorithme puis le code qui permet d'afficher la table de multiplication jusqu'à 12\*12 selon le format ci-dessous :

x	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	2	4	6	8	10	12	14	16	18	20	22	24
3	0	3	6	9	12	15	18	21	24	27	30	33	36
4	0	4	8	12	16	20	24	28	32	36	40	44	48
5	0	5	10	15	20	25	30	35	40	45	50	55	60
6	0	6	12	18	24	30	36	42	48	54	60	66	72
7	0	7	14	21	28	35	42	49	56	63	70	77	84
8	0	8	16	24	32	40	48	56	64	72	80	88	96
9	0	9	18	27	36	45	54	63	72	81	90	99	108
10	0	10	20	30	40	50	60	70	80	90	100	110	120
11	0	11	22	33	44	55	66	77	88	99	110	121	132
12	0	12	24	36	48	60	72	84	96	108	120	132	144

## F - C'est Noël

Proposer les codes qui permettent de réaliser les dessins suivants à l'aide des caractères « \* », espace et retour à la ligne en fonction d'un nombre de lignes N.  
exemple pour N=7

a- \*  
 \*\*  
 \*\*\*  
 \*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*

e-  
 \*                   \*  
 \*                   \*  
 \*                   \*  
 \*                   \*  
 \*                   \*  
 \*                   \*  
 \*                   \*

b- \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*

c-                   \*  
                   \*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*

f-                   0  
                   \*  
                   \*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*  
                   \*\*\*\*\*

d - \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*

G - Exo

Ecrire l'algorithme qui calcule la moyenne de x notes d'élèves à un devoir ainsi que la note la plus basse et la plus haute saisie.

H - Exo

Ecrire l'algorithme qui calcule et affiche le factoriel d'un nombre saisi au clavier.

I - Exo

Ecrire l'algorithme du jeu suivant : Le jeu se déroule avec 2 joueurs.

Le premier joueur saisit un mot à l'abri du regard de l'autre.

Puis, le deuxième joueur doit deviner quel est ce mot et le saisir.

Il a droit à 3 essais.

A chaque essai, le programme lui indique s'il a trouvé le mot ou bien le cas échéant, quelles sont les bonnes lettres aux bonnes places parmi celles qu'il a saisies

---

Exemple : si le mot à trouver est « carpette » et que le joueur a saisi « cratères », le programme lui affiche c- - - e - - -.

Si au bout de 3 essais, le joueur n'a pas trouvé le mot, un message lui indique quel était le mot à trouver.

#### J - Exo

Ecrire l'algorithme qui traduit un mot de n'importe quelle longueur saisi au clavier en codes numériques ascii.

Exemple : si l'utilisateur saisit le mot « jour », l'algorithme affiche le message suivant :

le code secret du mot « jour » est de 106 111 117 114.

Ces nombres sont les codes ascii des symboles j, o, u, r.

#### K - Exo

Dans un but statistique, on veut établir des calculs de la consommation d'un groupe de 100 abonnés à l'EDF, le dernier et l'avant dernier relevé de compteur de chacun d'eux étant saisi.

On sait d'autre part que la tarification se fait par tranches :

si la quantité d'électricité est inférieure à 100 kWh, le prix du kWh est de 0,09 euros;

si la quantité d'électricité est supérieure à 100 kWh, les 100 premiers kWh sont à 0,09 euros le kWh et au delà le prix du kWh est de 0,05 euros.

Le coût forfaitaire de la location du compteur est de 6 euros hors taxes.

Ecrire l'algorithme qui calcule et affiche le pourcentage d'abonnés faisant partie de la 1ère tranche, de la deuxième tranche ainsi que la consommation moyenne de chaque tranche et du groupe complet d'abonnés.

#### L - Exo

Un nombre est dit « parfait » s'il est égal à la somme de tous ces diviseurs sur lui même. Ecrire un algorithme qui indique si un nombre saisi au clavier est parfait ou pas.

#### M - Exo

Le but du jeu est de découvrir un nombre choisi aléatoirement par l'ordinateur compris entre 1 et 100. Lorsque le joueur saisit un nombre, l'ordinateur lui indique si ce nombre est plus petit ou plus grand que le nombre à découvrir. L'ordinateur affichera le nombre de coups qui ont été nécessaires pour découvrir le nombre mystère. On fixera le nombre d'essais à 5.

---

## 9 - Exercices sur les tableaux

1. Ecrire un programme qui saisit un entier au clavier et qui recherche si cet entier appartient au tableau (réponse de type oui/non) et affiche le rang où se trouve cet entier.
2. Ecrire un programme qui saisit deux indices et échange les valeurs contenues dans le tableau à ces deux indices. Le programme affichera le contenu du tableau avant et après cette transformation.
3. Ecrire un programme qui recherche et affiche le plus grand élément d'un tableau.
4. A partir d'une liste de nombres rangés de manière aléatoire dans un tableau, proposer le code qui classe ces mêmes nombres par ordre croissant dans le tableau.
5. voici une suite de nombre dans un tableau

-20	0	20	35	20	22	37	50	35	37
-----	---	----	----	----	----	----	----	----	----

Proposer le code qui inverse l'ordre de ce tableau comme le tableau suivant :

37	35	30	27	22	20	15	10	0	-20
----	----	----	----	----	----	----	----	---	-----

5. A partir des valeurs contenues dans deux tableaux. Proposer le code qui place dans un troisième tableau la valeur maximale trouvée entre deux valeurs de même rang des deux autres tableaux.

Exemple:

tableau 1

-20	0	20	35	20	2	37	50	-3	7
-----	---	----	----	----	---	----	----	----	---

tableau 2

-30	-2	10	15	20	17	27	30	35	20
-----	----	----	----	----	----	----	----	----	----

tableau 3 : max(tableau1, tableau2)

-30	0	20	35	20	17	37	50	35	20
-----	---	----	----	----	----	----	----	----	----

6. Ecrire un programme en langage C qui lit les dimensions L et C d'un tableau T à deux dimensions du type int (dimensions maximales : 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

---

## 10 - Exercices sur les « chaînes de caractères »

1. Ecrire le programme qui demande à l'utilisateur d'entrer un nombre entier compris entre 1 et 7 (seulement) et renvoie le nom du jour de la semaine en utilisant le formalisme tableau.  
On prendra comme convention le nombre 1 pour la chaîne de caractère « Lundi », 2 pour « Mardi », etc...
2. Ecrivez un programme qui prend une chaîne en majuscules et qui la convertit en minuscules (solution).  
le code ASCII des caractères alphabétiques en minuscules est compris entre 97 (a) et 122 (z)  
le code ASCII des caractères alphabétiques en majuscules est compris entre 65 (A) et 90 (Z)
3. Ecrire un programme qui demande un mot à un utilisateur et un autre utilisateur essaie de deviner ce mot en proposant des lettres. Le programme comparera les lettres à celles présentes dans le mot de départ et affichera les lettres trouvées à leur place. Les lettres inconnues seront remplacées par le symbole « \_ ». Le programme affichera le nombre de propositions fournies par l'utilisateur lorsque le mot est trouvé.
4. Ecrire le programme qui compte le nombre d'occurrence de chaque lettre présente dans une phrase qu'un utilisateur fournira.
- 4bis. Ecrivez un programme qui concatène deux chaînes de caractères.
5. Ecrire un programme qui calcule la longueur d'une chaîne de caractère. Donner le code qui permet de comparer deux chaînes de caractères de taille quelconque ne dépassant pas 255 caractères.  
Chaque mot d'une phrase doit avoir la même taille. Cette taille sera fixée par la taille du plus grand mot trouvé dans la phrase. Les mots qui sont plus courts seront complétés par autant de symbole « \_ » que nécessaire dans la nouvelle phrase.
6. Ecrire un programme qui supprime tous les caractères 'e' présents dans une phrase de 255 caractères maximum.
7. Manipulations de chaîne de caractères.
  - Définir un tableau de 100 caractères
  - Lire un texte et l'afficher
  - Sachant qu'une chaîne de caractère se termine par \0, quelle est la longueur du texte saisi?
  - Créer un tableau contenant les voyelles minuscules et majuscules. Combien y a-t-il de voyelles dans le texte saisi?
  - Recopier la chaîne de caractère d'un tableau dans un autre tableau.
  - Ecrire la chaîne de caractère saisi en majuscules exclusivement.
  - Réécrire le texte saisi en verlan.
  - Proposer un algorithme qui détermine si une chaîne de caractères est un palindrome
  - Proposer un algorithme qui permet de savoir si un caractère apparaît plusieurs fois dans un texte et de compter ses occurrences.



---

## 11 - Exercices sur les structures de données

1 - Réaliser un programme établissant une facture pouvant porter sur plusieurs articles. Pour chaque article à facturer, l'utilisateur ne fournira que la quantité et un numéro de code à partir duquel le programme devra retrouver à l'fois le libellé et le prix unitaire et afficher le montant total des articles achetés.

Le programme devra refuser les codes inexistants. A la fin, il affichera un récapitulatif tenant lieu de facture.

A) on commencera par définir une structure article.

B) on vérifiera les codes tapés au clavier pour vérifier si ils appartiennent à la liste de produits disponibles

C) on stockera ainsi les produits à acheter ainsi que leur quantité dans un tableau de structures (de taille 6 cases maximum)

D) on affichera les produits à acheter ainsi que leur prix sous forme de facture, par exemple sous le format suivant :

code	Désignation	Prix unitaire	quantité	Prix Total
11	gaufrier	26.80	2	53.60
16	grille-pain	19.95	1	19.95
25	centrifugeuse	37.00	10	370.00
26	four a raclette	29.95	20	599.00
<b>Montant :</b>				1042.55

Voici la liste des articles à intégrer dans le code :

code	Désignation	Prix
11	gaufrier	26.80
14	cafetière	23.50
16	grille-pain	19.95
19	balance de ménage	27.80
25	centrifugeuse	37.00
26	four a raclette	29.95

2 - Les nombres complexes peuvent être considérés comme un couple de réels : le nombre complexe  $3+4i$  est représenté par le couple (3,4).

Nous allons implanter l'arithmétique des complexes et pour ce faire, il nous faut définir un type `Complexe`. Nous pourrions utiliser un tableau de deux entiers mais nous allons plutôt utiliser une structure.

Pour implanter notre arithmétique, il nous faut coder :

- une fonction d'affichage d'un complexe : la fonction `PrintComplexe` prend en argument une variable de type `Complexe` et affiche un complexe ;
- l'addition de deux complexes : la fonction `addComplexe` prend en argument deux variables de types `Complexe` et retourne une variable de type `Complexe` codant la somme des deux arguments ;
- le produit de deux complexes : la fonction `mulComplexe` prend en argument deux variables de types `Complexe` et retourne une variable de type `Complexe` codant le produit des deux arguments ;
- le quotient de deux complexes : la fonction `quoComplexe` prend en argument deux variables de types `Complexe` et retourne une variable de type `Complexe` codant le quotient du premier argument par le second.

3 - On souhaite gérer une base de données d'inscriptions pour l'organisation d'un congrès qui dure une journée. Les organisateurs proposent aux participants de s'inscrire pour des repas, ainsi que pour l'hébergement en hôtel.

Un participant peut s'inscrire indépendamment aux 2 repas proposés : déjeuner (15 euros) et/ou dîner (35 euros) ou aucun. Il n'est pas obligé de prendre un hôtel. S'il en prend un, il peut choisir parmi 2 types d'hôtels différents : 2 étoiles (75 euros) ou 3 étoiles (100 euros).

Un participant peut venir accompagné de son conjoint. Dans ce cas, la réservation d'hôtel est identique mais lorsqu'un repas est sélectionné alors il faut en compter 2.

a) Créer un nouveau type *Participant* qui inclut son nom (chaîne de caractères : tableau de 20 caractères), son prénom (chaîne de caractères : tableau de 30 caractères), ainsi que toutes les autres informations nécessaires à son inscription selon les critères définis ci- dessus. On privilégiera une structure contenant un nombre minimal de champs.

b) Créer un nouveau type *TabPart* qui est un tableau de 100 éléments de type *Participant*.

c) Créer une fonction *Nb2Etoiles* qui, pour un argument de type *TabPart* donné, affiche le nom et le prénom des personnes qui ont choisi de réserver un hôtel 2 étoiles. On pourra utiliser la fonction `printf` avec pour format `%s` pour l'affichage des chaînes de caractères nom et prénom.

d) Créer une fonction *NbDej* qui, pour un argument de type *TabPart* donné, retourne le nombre de déjeuners à prévoir.

e) Créer une fonction *Montant* qui calcule, pour un *Participant* donné en argument, le montant de sa facture.

#### 4 - Point 2D

– Définir un type *point2D* qui représente les points du plan.

– Écrire une fonction qui calcule la distance entre deux points.

– Définir un type *segment*.

– Écrire une fonction qui renvoie le milieu d'un segment.

– Définir un type *vecteur*

– Écrire une fonction qui renvoie le vecteur associé à deux points.

– Définir un type *droite* donné par un point de la droite et un vecteur directeur.

– Écrire une fonction qui calcule l'intersection de deux droites.

---

– Écrire une fonction qui calcule la droite perpendiculaire à une droite et passant par un point donné.

#### 5 - Pays et capitales

– Définir le type pays qui représente les pays limitrophes de la France. – Écrire une fonction qui donne la capitale de chacun de ces pays.

#### 6 - Jour de la semaine

– Définir le type semaine qui représente chaque jour de la semaine.

– Écrire une fonction qui teste si un jour de la semaine est un jour du week-end.

## **11 - Exercices avec le formalisme « pointeur »**

1. Soit le tableau  $t$  déclaré comme suit :

`Float t[3][4] ;`

Donner le code qui permet de faire la somme de chaque élément du tableau puis la moyenne du tableau :

- En utilisant le formalisme tableau
- En utilisant le formalisme pointeur

2. En utilisant le formalisme pointeur, donner le code qui permet de classer par ordre croissant les éléments d'un tableau.

3. Donner le code qui permet de classer par ordre alphabétique une suite de chaîne de caractère.

Par exemple à partir de la suite « Lundi », « Mardi », « Mercredi », « Jeudi », « Vendredi », « Samedi », « Dimanche », on les classe par ordre alphabétique : « Dimanche », « Jeudi », « Lundi », « Mardi », « Mercredi », « Samedi » et « Vendredi ».

4. Donner le code qui calcule le produit de 2 matrices de nombres réels de taille quelconque. On vérifiera si le produit est réalisable (le nombre de colonnes de la matrice A est le nombre de lignes de la matrice B).

---

## 12 - Exercices sur les fonctions et sous programmes

1. Soit un nombre tiré aléatoirement (fonction rand()), écrire un programme qui permettra à un utilisateur de deviner ce nombre en indiquant si les propositions de l'utilisateur sont plus grand, plus petit ou exact. On fixera un certain nombre d'essai possible (10). pA l'issu du nombre d'essais infructueux, la partie recommence. Pour sortir du programme on appuiera sur la touche A.

2. Quels sont les affichages liés aux programmes suivants

### Programme 1

```
#include <stdio.h>
int fct(int);

int main(int argc, char *argv[])
{
    int n,p=5;
    n=fct(p);
    printf("p=%d, n=%d\n",p,n);
}

int fct(int r)
{
    return 2*r;
}
```

### Programme 2

```
#include<stdio.h>
int n=10,q=2;
int fct(int);
void f(void);

int main(int argc, char *argv[])
{
    int n=0,p=5;
    n=fct(p);
    printf("A: dans main, n=%d, p=%d, q=%d\n",n,p,q);
    f();
}

int fct(int p)
{
    int q;
    q=2*p+n;
    printf("B: dans fct, n=%d, p=%d, q=%d\n",n,p,q);
    return q;
}
void f(void)
```

---

```

{
    int p=q*n;
    printf("C : dans f, n=%d, p=%d, q=%d",n,p,q);
}

```

3. Réaliser le programme qui permettra de calculer la surface d'un cercle avec deux sous programmes :

- l'un permettant de saisir le rayon du cercle
- l'autre effectuant le calcul

L'affichage se fera dans le programme principal.

4. Proposer l'algorithme et le code en langage C qui permettra de réaliser une calculatrice à 4 opérations. On écrira une fonction qui acceptera en paramètres les 2 opérandes et un caractère qui permettra de choisir le type d'opération à effectuer (+, -, \*, /).

5. Tri par permutation ou tri « à bulle »

Ecrire une fonction réalisant le tri par valeurs croissantes d'un tableau d'entiers<sup>2</sup>, en utilisant l'algorithme du tri par permutation simple qui se définit ainsi :

On parcourt l'ensemble du tableau, depuis sa fin jusqu'à son début, en comparant deux éléments consécutifs, en les inversant s'ils sont mal classés. On se retrouve ainsi avec le plus petit élément placé en tête du tableau.

On renouvelle l'opération (dite « passe ») avec les n-1 éléments restants, puis avec les n-2 éléments restants et ainsi de suite jusqu'à ce que :

- soit l'avant dernier élément ait été classé (le dernier étant obligatoirement à sa place).
- soit qu'aucune permutation n'ait eu lieu pendant la dernière passe

Exemple d'interaction avec le programme:

Combien de valeurs à trier : 6

Donner les valeurs à trier : 2 8 4 7 0 8

Algorithme

```

2 8 4 7 0 8 //valeurs à trier
2 8 4 0 7 8 // échange de 0 et 7
2 8 0 4 7 8 // échange de 0 et 4
2 0 8 4 7 8 // échange de 0 et 8
0 2 8 4 7 8 // échange de 0 et 2
0 2 4 8 7 8 // échange de 4 et 8
0 2 4 7 8 8 // échange de 8 et 7

```

On prévoira en argument à la fonction de tri :

- l'adresse du tableau à trier,
- son nombre d'éléments,
- un indicateur précisant si l'on souhaite que la fonction affiche les valeurs du tableau après chaque permutation (0 pour oui et 1 pour non).

---

<sup>2</sup> n représente le nombre d'éléments dans le tableau