

Ressource

Cahier d'exercices v 1.51



(C) Achille Braquetaire



Lycée E. Branly BTS SNEC Année 2021-2022

Consignes :



- Créez un répertoire *CahierExercices*
- Sauvegardez tous les fichiers sources dans le répertoire créé
- Le fichier source de l'exercice N°1 se nommera exercice01.c (ne pas oublier le 0), pour l'exercice 15 : exercice15.c
- Dans le cas où il y a plusieurs étapes, créez un répertoire portant le nom de l'exercice
- Utilisez soit codeblocks ou mu editor (voir avec l'enseignant) soit http://www.tutorialspoint.com/compile_c_online.php
- <https://www.codechef.com/ide> s'affiche mieux au sein du lycée
- Par préférence :
 - Langage C : codeblock ou <https://www.onlinegdb.com/>
 - Python : mu editor ou <https://replit.com/languages/python3>
- <http://highlight.hohli.com/> permet une coloration syntaxique

Table des matières

I.	En rapport avec le memento	7
	Exercice N°1. Premier programme *	7
II.	Programmes syntaxe de base	7
	Exercice N°2. Bonjour Monsieur1ts *	7
	Exercice N°3. Carré d'un nombre *	7
	Exercice N°4. Calculer la surface d'un cercle à partir du rayon1ts *	7
	Exercice N°5. Moyenne *	7
	Exercice N°6. Volume d'une sphère *	8
	Exercice N°7. Conversion radian degré *	8
	Exercice N°8. Divisible par 3 (voir memento) **	8
	Exercice N°9. Prix et nombre d'articles **	8
	Exercice N°10. Moyenne arithmétique pondérée **	8
	Exercice N°11. Où placer les déclarations de variables ? ***	8
	Exercice N°12. Modulo 2π ***	9
	Exercice N°13. Conversion Mo/s en bits/s *	9
	Exercice N°14. dpi d'une image *	9
	Exercice N°15. taille d'une image **	9
III.	Les conditionnels	10
	Exercice N°16. Plus grand que 10 ou plus petit1ts *	10
	Exercice N°17. Quel est le plus grand [condition](*) 1ts *	10
	Exercice N°18. Valeur absolue [condition] *	10
	Exercice N°19. Même exercice mais avec plus grand et plus petit1ts *	10



Exercice N°20.	Conversion de notes chiffrées en lettres *	10
Exercice N°21.	Remise 5% *	11
Exercice N°22.	Remise 5 et 10 % **	11
Exercice N°23.	Catégorie d'âges **	11
Exercice N°24.	Poids d'une lettre1ts **	11
Exercice N°25.	Lendemain d'une journée **	11
Exercice N°26.	Convertir les octets en Mo, ko et octets **	11
Exercice N°27.	Suite ***	12
Exercice N°28.	Année bissextiles ***	12
Exercice N°29.	Trier trois nombres ***	12
Exercice N°30.	Une minute après ***	12
Exercice N°31.	Compagnie d'assurance ***	13
Exercice N°32.	impôt sur le revenu **	13
Exercice N°33.	RS232 ***	13
IV.	Les boucles	14
Exercice N°34.	Tant que >10 1ts *	14
Exercice N°35.	Afficher 1000 fois "Bonjour"1st *	14
Exercice N°36.	Moyenne des températures de la semaine1ts *	14
Exercice N°37.	Jour du mois de Janvier *	14
Exercice N°38.	Somme des 100 premiers entiers [boucle] *	15
Exercice N°39.	Nombre mystère **	15
Exercice N°40.	Nombre mystère bis**	15
Exercice N°41.	Réécriture de for **	15
Exercice N°42.	Factorielle1ts **	15
Exercice N°43.	Moyenne nombres saisis 1ts **	15
Exercice N°44.	Triangle d'étoiles **	16
Exercice N°45.	Les trois boucles pour, tant que faire, faire tant que **	16
Exercice N°46.	Nombre mystérieux **	16
Exercice N°47.	Vérification d'une note **	16
Exercice N°48.	Radio FM ***	16
Exercice N°49.	Radio FM - mémorisation - ***	16
Exercice N°50.	Imbriquer deux boucles heures :minutes de la journée ***	17
Exercice N°51.	Bonjour n fois *	17



Exercice N°52. Moyenne de notes version 2 *** 17

Exercice N°53. Table de multiplication version 2 *** 17

Exercice N°54. Adresse IP **** 18

Exercice N°55. Adresse ip de classe C * 20

Exercice N°56. Adresse ip de classe B ** 20

V. Les tableaux 20

Exercice N°57. Tableau : initialisation à zéro * 20

Exercice N°58. Eléments > à 10 * 20

Exercice N°59. Pairs et impairs ** 20

Exercice N°60. Le plus grand ** 21

Exercice N°61. Tableaux à 2 dimensions *** 21

Exercice N°62. Le plus grand/petit, la moyenne ** 21

Exercice N°63. Répertoire téléphonique d'un hôtel *** 21

Exercice N°64. Répertoire téléphonique d'un hôtel (niveau II) *** 22

Exercice N°65. Manipulation de tableaux *** 22

Exercice N°66. Recherche d'un élément *** 22

Exercice N°67. Tableau à deux dimensions *** 22

VI. Les fonctions 22

 a. Ressemblant aux exercices précédents 22

Exercice N°68. Afficher 5 fois * 22

Exercice N°69. Afficher n fois * 23

Exercice N°70. Conversion ko en bits * 23

 b. Un peu de math 23

Exercice N°71. Factorielle * 23

Exercice N°72. Table de multiplication * 23

Exercice N°73. Table de multiplication v2 * 23

Exercice N°74. Est pair * 24

Exercice N°75. Est_impair * 24

Exercice N°76. Est_premier 24

Exercice N°77. Premiers suite 24

Exercice N°78. Liste chiffres 24

Exercice N°79. Boucle inverse * 24

Exercice N°80. DixNombres * 24





Exercice N°81.	La fonction MoyenneTroisNotes *	24
Exercice N°82.	Afficher n fois un mot **	25
Exercice N°83.	Occurrence **	25
Exercice N°84.	Conversion ***	25
Exercice N°85.	Fonction maximum	25
Exercice N°86.	Fonction : taille d'une image **	25
c.	En lien avec les tableaux	26
Exercice N°87.	Fonction moyenne **	26
Exercice N°88.	Fonction minimum *	26
Exercice N°89.	Fonction position_minimum **	26
Exercice N°90.	Fonction contient_valeur *	26
Exercice N°91.	Fonction position_valeur *	26
Exercice N°92.	Médiane **	26
Exercice N°93.	Valeurs_sup ***	26
Exercice N°94.	Valeur_sup_moyenne **	27
Exercice N°95.	Reverse **	27
Exercice N°96.	Est_croissante **	27
Exercice N°97.	Est_decroissante **	27
Exercice N°98.	plus_longue_suite_croissante() ***	27
VII.	les chaînes de caractères	27
a.	En langage C	27
Exercice N°99.	Utilisation des fonctions de manipulations de chaînes de caractères ***	28
b.	Concernant les dates avec des fonctions	28
Exercice N°100.	Fonction mois *	28
Exercice N°101.	Heure_chaine **	28
Exercice N°102.	Heure_chaine **	28
Exercice N°103.	AjouterAnnees **	29
Exercice N°104.	AjouterMois **	29
Exercice N°105.	AjouterJour ***	29
Exercice N°106.	AjouterHeures ***	29
Exercice N°107.	AjouterMinutes ***	29
Exercice N°108.	AjouterSecondes ***	29
VIII.	Les structs	29



Exercice N°109. Structure eleve *** 29

Exercice N°110. HeuresMnsSec *** 30

IX. Champion des maths **** 30

Exercice N°111. Page d'accueil 30

Exercice N°112. Choix multiples 30

Exercice N°113. Addition 30

Exercice N°114. multiplication 31

Exercice N°115. Soustraction 31

Exercice N°116. Sortie du jeu 31

Exercice N°117. Utiliser un switch case 31

Exercice N°118. Gestion des points 31

Exercice N°119. Tables de multiplication 31

Exercice N°120. Tables de multiplication (suite) 32

X. Autres problèmes niveau **** 32

Exercice N°121. Le coffre-fort **** 32

Exercice N°122. Problème : parité d'un nombre binaire **** 32

Exercice N°123. Assurance **** 34





I. EN RAPPORT AVEC LE MEMENTO

Exercice N°1. Premier programme *

1. Déclarer les variables j (entier, valeur 3), $diametre$ (réel, valeur 3.14), $prenom$ (votre nom)
2. Les afficher (l'affichage doit ressembler à `j=3 diametre=3.14 prenom=toto`)
3. Les demander à l'utilisateur puis les afficher à nouveau
4. Si le prénom entré est « toto », affichez « Coucou c'est toto »

II. PROGRAMMES SYNTAXE DE BASE

Exercice N°2. Bonjour Monsieur1ts *

Ecrire un algorithme qui vous demande de saisir votre nom puis votre prénom et qui affiche ensuite « Bonjour » suivi du nom complet

Variable:

nom : chaînes de caractères

Début

Afficher "quel est votre prénom ?"

Saisir nom

Afficher "Bonjour " , nom

fin

Exercice N°3. Carré d'un nombre *

Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.

Exercice N°4. Calculer la surface d'un cercle à partir du rayon1ts *

Entrée : Saisie du rayon

Sortie : Affichage de la surface du disque

Traitement à réaliser : $Surface = \pi * Rayon^2$

Exercice N°5. Moyenne *

Écrire un algorithme qui calcule la moyenne arithmétique de trois nombres donnés.



Exercice N°6. **Volume d'une sphère ***

Le volume d'une sphère est $\frac{4}{3} * \pi * r^3$: $\frac{4}{3} * \pi * r^3$.

Demandez le rayon et affichez le volume (prendre Pi=3.14)

Exercice N°7. **Conversion radian degré ***

π radian = 180°

Pour convertir les degrés en radians : on multiplie la mesure de l'angle par π , puis on divise le résultat par 180°.

Demandez un angle en radian et l'afficher en degré

Exercice N°8. **Divisible par 3 (voir memento) ****

Demandez un nombre et affichez le reste de la division par 3

Exercice N°9. **Prix et nombre d'articles ****

Ecrire un programme qui lit le prix HT (hors taxe) d'un article, le nombre d'articles et le taux de TVA (Taxe sur la valeur ajoutée), et qui fournit le prix total TTC (Toutes Taxes Comprises) correspondant. Faire en sorte que des libellées apparaissent clairement. La TVA est de 20% (donc 0.2).

Formule : PrixTTC = PrixHT * TVA = **PrixHT * 0.2**

Exercice N°10. **Moyenne arithmétique pondérée ****

Même question pour une moyenne pondérée quand on donne les nombres et les coefficients de pondération.

Exercice N°11. **Où placer les déclarations de variables ? *****

La portée d'une variable est l'espace dans lequel elle est définie.

 Sans exécuter les programmes, donner leur affichage.

```
#include <stdio.h>
int compteur=10;
```

```
int main()
{
  int compteur=10;
  printf("compteur=%d \n",compteur);
  if (compteur == 10)
  {
    int compteur;
    printf("compteur=%d\n",compteur);
  }
  return 0;
}
```

```
#include <stdio.h>
int compteur=10;

int main()
{
  printf("compteur=%d \n",compteur);
  if (compteur == 10)
  {
    int compteur;
    compteur ++;
    printf("compteur=%d\n",compteur);
  }
  return 0;
}
```



Exercice N°12. Modulo 2π ***

Un angle modulo 2π veut dire que l'on enlève une ou plusieurs fois 2π (ou 360°). Par exemple :

- $370^\circ = 10^\circ$ (modulo 2π , 360)
- $361^\circ = 1^\circ$ (modulo 2π , 360)
- $765^\circ = 45^\circ$ ($+2*360^\circ$)
- ...

Modulo 360° (ou 2π) représente simplement le reste de la division entière par 360 .

Ecrire un programme qui demande un angle à l'utilisateur, puis qui l'affiche modulo 360° (ou 2π) et modulo π

Exercice N°13. Conversion Mo/s en bits/s *

Demandez la vitesse en Mo/s et le convertir en bits/s (prenez $1\text{Mo}=1000\text{ko}$)

Exercice N°14. dpi d'une image *

La résolution se caractérise par le nombre de pixels par unité de longueur. On parle de dpi (dot per inch) ou ppp (point par pouce).

On rappelle que $1'' = 2,54$ cm.

Par exemple : un écran 4/3 de $19''$ possède une taille de $38,61$ cm x $28,96$ cm et respectivement une définition de 1920×1440 pixels.

Le dpi = $(1920/38.61)*2.54=126.3$

Demandez la taille en hauteur de l'écran et la définition verticale en pixel puis affichez le dpi

Exercice N°15. taille d'une image **

La taille d'une image peut être calculé ainsi : nombre de pixels en largeur * : nombre de pixels en hauteur*taille d'un pixel

Par exemple



La taille est : $300*200*24$ bits/pixels = 1440000 bits = $1440000/8=180000$ octets = $180000/1024=175.8$ ko
Demandez le nombre de pixels en largeur, en hauteur et le nombre de bits/pixels puis affichez la taille en ko.



III. LES CONDITIONNELS

Exercice N°16. Plus grand que 10 ou plus petit1ts *

Demandez un entier à l'utilisateur et si ce nombre > 10 "Le nombre est plus grand que 10"

Exercice N°17. Quel est le plus grand [condition](*) 1ts *

Demander deux nombres entiers puis les afficher les deux nombres dans l'ordre croissant

Exercice N°18. Valeur absolue [condition] *

Le programme demande un entier, positif ou négatif, et affiche sa valeur absolue.

Exercice N°19. Même exercice mais avec plus grand et plus petit1ts *

Demandez un entier à l'utilisateur et si ce nombre > 10 "Le nombre est plus grand que 10" et "Le nombre est plus petit que 10" dans le cas contraire

Si Nombre > 10

Alors afficher " Le nombre est plus grand que 10"

Sinon afficher " Le nombre est plus petit que 10"

Fin SI

Exercice N°20. Conversion de notes chiffrées en lettres *

Un professeur souhaite n'attribuer que 5 types de notes.

Pour cela, il désire convertir les notes chiffrées en lettres A, B, C, D, et E, d'après les règles suivantes :

A pour toutes les notes comprises entre 20 et 17 inclus

B pour toutes les notes comprises entre 16 et 13 inclus

C pour toutes les notes comprises entre 12 et 9 inclus

D pour toutes les notes comprises entre 8 et 4 inclus

E pour toutes les notes comprises entre 3 et 0 inclus

Voici un exemple d'utilisation :

Quelle est la note (entre 0 et 20) ? 15

Cette note correspond à un B.

Quelle est la note (entre 0 et 20) ? 16

Cette note correspond à un B.

Quelle est la note (entre 0 et 20) ? 17

Cette note correspond à un A.

Quelle est la note (entre 0 et 20) ? 21

Cette note est hors limite.



Quelle est la note (entre 0 et 20) ? 12

Cette note correspond à un C

- Ecrire un programme de conversion pour une note entière. L'affichage doit être impérativement le même que celui ci-dessus.

Exercice N°21. Remise 5% *

Vous demandez le prix d'un article et le nombre désiré, si total est supérieur à 1500 €, alors la remise est de 5%.

Exercice N°22. Remise 5 et 10 % **

Même question, mais si le total est > à 3000 €, la remise est de 10 % au lieu des 5%

Exercice N°23. Catégorie d'âges **

Ecrire un algorithme qui demande l'âge d'un enfant puis l'informe de sa catégorie : "Poussin" de 6 à 7 ans, "Pupille" de 8 à 9 ans, "Minime" de 10 à 11 ans et "Cadet" après 12 ans.

Exercice N°24. Poids d'une lettre 1ts **

Écrire un programme qui demande le poids de la lettre puis affiche le tarif du timbre :

ECOPLI		Service économique d'envoi de lettres et petits objets jusqu'à 3 cm d'épaisseur	
POIDS JUSQU'À		TARIFS NETS	
20 g		0,66€	
50 g		1,05€	
100 g		1,45€	
250 g		2,50€	

Exercice N°25. Lendemain d'une journée **

Écrire un programme permettant de calculer le lendemain d'une journée donnée (jour, mois, année). On ne tiendra pas compte des années bissextiles et le mois de février

Exercice N°26. Convertir les octets en Mo, ko et octets **

Le but de cet exercice est de convertir un nombre d'octets en Mo, ko et le reste en octets.

On prendra 1Mo=1024ko et 1Ko=1024 octets

Exemples :

Quelle est la taille en octets ? : 1024

1024=0Mo+1ko+0octets



BTS SN - EC
- Langage C -
Cahier d'exercices



Quelle est la taille en octets ? : 1026
1026=0Mo+1ko+2octets

Quelle est la taille en octets ? : 1048576
1048576=1Mo+0ko+0octets

Quelle est la taille en octets ? : 2000000
2000000=1Mo+929ko+128octets

Une autre version est demandée :

Au lieu d'avoir cet affichage :

Quelle est la taille en octets ? : 1024
1024=0Mo+1ko+0octets

Au lieu d'avoir cet affichage :

Quelle est la taille en octets ? : 1024
1024=1ko

Exercice N°27. Suite ***

Les nombres étant entiers, utiliser maintenant une structure à choix multiple.

Exercice N°28. Année bissextiles ***

(wikipedia)

Depuis l'ajustement du calendrier grégorien, l'année sera bissextile (elle aura 366 jours) :

- si l'année est divisible par 4 et non divisible par 100, ou
- si l'année est divisible par 400.

Demander l'année à l'utilisateur est lui afficher si elle est bissextile ou non.

Exercice N°29. Trier trois nombres ***

Demandez trois nombres, puis les afficher dans l'ordre croissant.

Exercice N°30. Une minute après ***

Écrire, avec des comparaisons, un algorithme qui lira au clavier les heures et les minutes, et qui affichera l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 20 puis 33, l'algorithme doit répondre :

"Dans une minute il sera 20 heure(s) 34 minute(s).

NB : on suppose que l'utilisateur entre une heure valide.



Exercice N°31. Compagnie d'assurance ***

Une compagnie d'assurance automobile doit appliquer une surprime suivant l'âge, le sexe et le nombre d'années de permis de la personne.

Les hommes de plus de 22 ans payent la surprime.

Les femmes entre 20 et 30 ans payent la surprime.

Les personnes ayant plus de 5 années de permis ne payent pas la surprime.

Écrire un algorithme qui détermine si la personne doit payer ou non la surprime.

Écrire le programme en Langage C équivalent.

Exercice N°32. impôt sur le revenu **

Ecrivez un algorithme qui interroge un habitant d'un pays lointain, sur son âge et son sexe, et lui annonce s'il doit payer l'impôt, sachant que :

- les habitants de 20 ans et plus paient l'impôt,
- les habitantes de plus de 18 ans et de moins de 35 paient l'impôt,
- les autres sont dispensés d'impôt.

Exercice N°33. RS232 ***

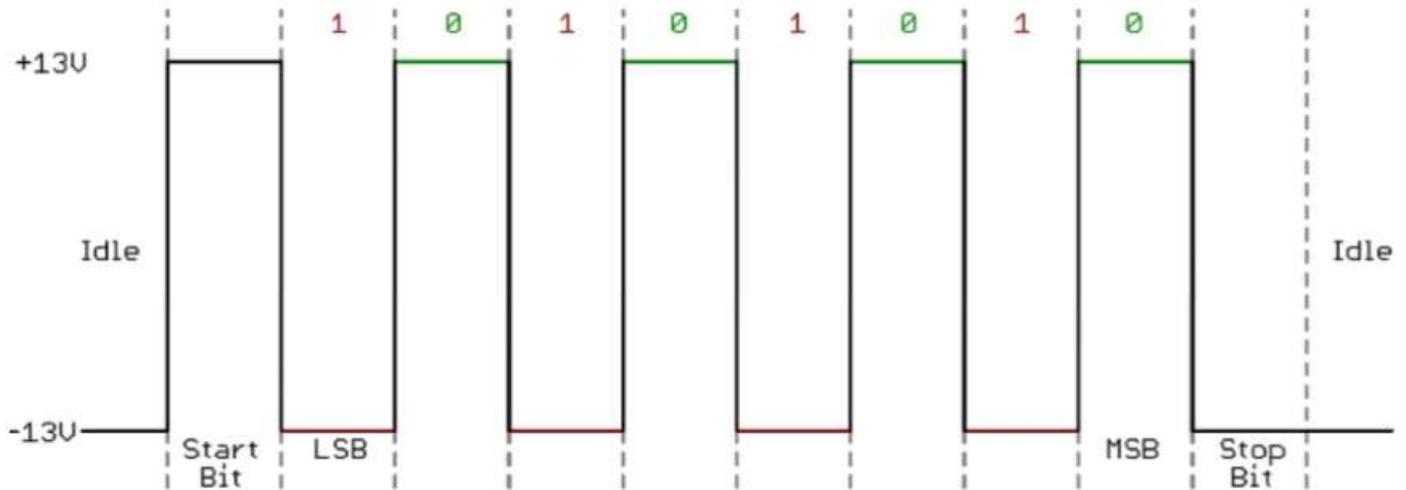
La caractéristique du RSR232 est

- La vitesse de transmission - bits par seconde - : 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 et 115200 bauds.
- Bits de données - longueur des mots - : 7 bits ou 8 bits (généralement du code ASCII).
- Parité : pair, impair ou pas de parité. Dans le cas d'un parité paire (even) : le bit ajouté à la donnée est positionné de telle façon que le nombre d'états « 1 » soit pair sur l'ensemble « donnée + bit de parité ».
- Bit de stop : après la transmission du mot, la ligne est positionnée au repos (à l'état logique « 1 ») pendant 1 ou 2 périodes d'horloge, selon le nombre de bits de stop.
- Contrôle de flux : aucun ou
 - Le protocole logiciel XOn -XOff : envoi de XOff (caractère ASCII 17) vers l'émetteur pour arrêter la transmission, puis de Xon (caractère 19) pour reprendre la transmission.
 - Le protocole matériel RTS-CTS : nécessite l'utilisation de deux fils supplémentaires dans le câble (liaison 5 fils). La broche RTS (Ready To Send - prêt à émettre) du premier équipement doit être reliée à la broche CTS (Clear To Send - demande d'arrêt)

Demandez le nombre de bits de données (7 ou 8), celui de bits de stop (1 ou 2) et affichez le nombre total de bits à envoyer en une seule fois. Il faudra vérifier la saisie de l'utilisateur (bits de données 7 ou 8, bits de stop 1 ou 2)



Par exemple pour



Il y a 10 bits.

IV. LES BOUCLES

Exercice N°34. Tant que >10 1ts *

Demander à l'utilisateur un nombre tant que ce dernier est >10 avec la boucle tant que

Exercice N°35. Afficher 1000 fois "Bonjour"1st *

Afficher 1000 fois "Bonjour"

Exercice N°36. Moyenne des températures de la semaine1ts *

Écrire un programme qui demande les températures de 7 jours, calcule puis affiche la température moyenne de la semaine.

Exercice N°37. Jour du mois de Janvier *

Écrire un programme qui affiche le calendrier du mois de janvier :

1 janvier
 2 janvier
 ...
 31 janvier



Exercice N°38. Somme des 100 premiers entiers [boucle] *

Ecrire un programme qui fait la somme des 100 premiers entiers.

Faites deux versions l'une avec la boucle *pour* et l'autre avec *tant que*

Exercice N°39. Nombre mystère **

L'utilisateur doit trouver un nombre tiré aléatoirement entre 0 et 10. Tant qu'il ne l'a pas trouvé le message « Non trouvé » s'affiche.

Exercice N°40. Nombre mystère bis**

Dans cette nouvelle version, programme indiquera si le nombre est plus grand ou plus petit, et donnera le nombre de tentatives avant de terminer.

Exercice N°41. Réécriture de for **

Soit le petit programme suivant :

```
#include <stdio.h>
main()
{
  int i, n, som ;
  som = 0 ;
  for (i = 0 ; i < 4 ; i++)
  { printf ("donnez un entier ") ;
    scanf ("%d", &n) ;
    som += n ;
  }
  printf ("Somme : %d\n", som) ;
}
```

Écrire un programme réalisant exactement la même chose, en employant, à la place de l'instruction *for* :

- une instruction *while* ,
- une instruction *do... while* .

Exercice N°42. Factorielle1ts **

Ecrire un algorithme qui demande un nombre de départ et qui calcule sa factorielle (exemple $5! = 1*2*3*4*5$)

Exercice N°43. Moyenne nombres saisis 1ts **

Ecrire un programme qui calcule la moyenne d'une série d'entiers. On demandera à l'utilisateur le nombre d'entiers qui sera saisi.



Exercice N°44. Triangle d'étoiles **

On souhaite afficher un triangle d'étoiles ayant 1 étoile au sommet et n étoiles à la base. Ci-dessous est représenté le résultat pour n = 5

*
 **

Implémenter un programme qui demande à l'utilisateur combien d'étoiles il désire, puis l'affiche.

Exercice N°45. Les trois boucles pour, tant que faire, faire tant que **

Écrire un programme qui affiche 10 fois « coucou ». Utiliser les trois boucles vues en cours : pour (for), tant que faire(while (...) {...}) , faire tant que ({...} while (...)).

Exercice N°46. Nombre mystérieux **

Soit un nombre mystérieux que vous initialiserez avec une valeur que vous choisirez, écrire un algorithme qui demande un nombre compris entre 1 et 20, jusqu'à ce que l'utilisateur trouve le nombre mystérieux. En cas de réponse supérieure, on fera apparaitre un message : Plus petit ! , et inversement, Plus grand ! si le nombre est inférieure.

Pour finir, si le nombre est trouvé, « Gagné » s'affichera et le programme s'arrêtera.

Exercice N°47. Vérification d'une note **

Le programme demande une note, il faut vérifier qu'elle soit bien comprise entre 0 et 20, dans le cas contraire il faut lui redemander la note tant que cette condition (comprise entre 0 et 20) n'est pas vérifiée.

Exercice N°48. Radio FM ***

Le programme va simuler l'affichage d'une fréquence de radio : l'utilisateur en appuyant sur '+' va augmenter la fréquence de 0,5 MHz et pour '-' diminué de 0.5MHz.

La page de fréquence est 87,5 - 108 MHz et lorsque l'utilisateur est arrivé à une des deux bornes (87.5 ou 108) il est bloqué. Le début de l'affichage se fera à 87.5 MHz

Exercice N°49. Radio FM – mémorisation – ***

Par la touche M suivi d'un chiffre (0 à 9), l'utilisateur peut mémoriser une fréquence. Le rappel de la radio mémorisée est effectué par R suivi du chiffre.

Par exemple :

101 MHz

M1

Radio mémorisée

101 MHz



+
101.5 MHz

+
102 MHz

R1

Rappel mémoire M1

101 MHz

Exercice N°50. *Imbriquer deux boucles heures :minutes de la journée*

Écrire un programme qui affiche l'heure :minutes de la journée. On débute à minuit 00 :00, 00 :01,...,10 :32,...15 :59 jusqu'à 23 :59

Il faut utiliser deux boucles imbriquées : une première pour les heures qui englobera celle des minutes

Exercice N°51. *Bonjour n fois **

Ecrire un programme, en utilisant une boucle, qui affiche

- bonjour 1 fois
- bonjour 2 fois
- bonjour 3 fois
- bonjour 4 fois
- bonjour 5 fois

Exercice N°52. *Moyenne de notes version 2 ****

Calculer la moyenne de notes fournies au clavier avec un dialogue de ce type :

- note 1 : 12
- note 2 : 15.25
- note 3 : 13.5
- note 4 : 8.75
- note 5 : -1
- moyenne de ces 4 notes : 12.37

Le nombre de notes n'est pas connu a priori et l'utilisateur peut en fournir autant qu'il le désire. Pour signaler qu'il a terminé, on convient qu'il fournira une note fictive négative. Celle-ci ne devra naturellement pas être prise en compte dans le calcul de la moyenne.

Exercice N°53. *Table de multiplication version 2 ****

Écrire un programme qui affiche la table de multiplication des nombres de 1 à 10, sous la forme suivante :

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40



5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Aide :

Il faut utiliser la tabulation \t dans le printf : `printf("%d\t")`

A la fin de la boucle de ligne, il faut ajoute un `printf("\n")`

Exercice N°54. Adresse IP ****

Soit le programme (<https://repl.it/LfSF/0>)

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#define OK 0
#define ERREUR 1

int main()
{
    char AdresseIP[125]="192.168.10.225";
    int octets[4];
    int valRetour=TrouverOctets(AdresseIP,octets);
    if (valRetour==ERREUR)
    {
        printf("Une erreur a été détectée");
    }
    else
    {
        // affichage octet par octet
        for (int i=0;i<=3;i++)
        {
            printf("L'octet N°%d est ",i);
            printf("%d\n",octets[i]);
        }
    }

    return(0);
}

// Fonction qui va aller récupérer les quatre octets

int TrouverOctets(char *chaine,int *octets)
{
    char *token;
    int nombreOctets=0;
    token = strtok(chaine, ".");// Recherche de la première occurrence
    do
    {
        *octets++=atoi(token);// converti une chaine en entier
        //printf("%d\n",*(octets-1));
        if((*octets-1)<0 || (*octets-1)>255) return ERREUR; // si pas entre 0.255
        token = strtok(NULL, ".");
    }
}
```

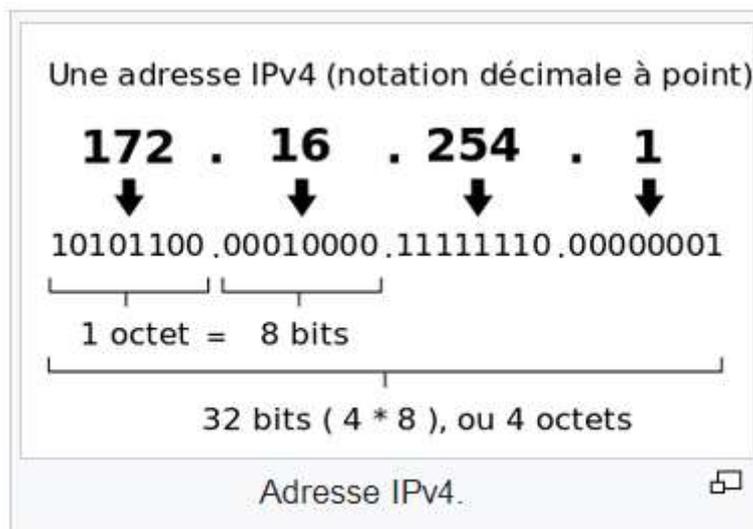


```

nombreOctets++;
}while( token != NULL ) ;
if (nombreOctets!=4) return ERREUR;// si 4 octets ne sont pas lus -> ERREUR
return OK;
}
  
```

Une adresse IP (avec IP pour Internet Protocol) est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque appareil connecté à un réseau informatique utilisant l'Internet Protocol.

Il existe des adresses IP de version 4 (sur 32 bits, soit 4 octets) et de version 6 (sur 128 bits, soit 16 octets). La version 4 est actuellement la plus utilisée : elle est généralement représentée en notation décimale avec quatre nombres compris entre 0 et 255, séparés par des points, ce qui donne par exemple : 172.16.254.1 (l'adresse qui figure sur l'image ci-dessous).



(src : wikipedia)

Ce programme analyse une chaîne de caractères contenant une adresse IP et range les quatre valeurs décimales (4 octets) dans un tableau (rôle de la fonction `TrouverOctets`)

 Testez ce programme

C'est une classe C, les adresses vont de 192.168.10.0 à 192.168.10.255

 Affichez toutes les adresses du réseau, il faut utiliser `octets[0]`, `octets[1]`, `octets[2]`

Les adresses suivantes ne sont pas disponibles :

- La première et la dernière
- Les quatre adresses 192.168.10.10, 192.168.10.20, 192.168.10.35 et 192.168.10.34

 Affichez toutes les adresses disponibles du réseau.

Maintenant nous allons utiliser une adresse réseau de classe B 172.16.0.0 dont les valeurs vont de :

172.16.0.0 à 172.168.0.255

Puis de 172.168.1.0 à 172.168.1.255

Puis de 172.168.2.0 à 172.168.2.255

...

Puis de 172.168.255.0 à 172.168.255.255



BTS SN - EC

- Langage C -

Cahier d'exercices



- Remplacez à la ligne 9 `char AdresseIP[125]="192.168.10.225";` par `char AdresseIP[125]="172.16.0.0";`
- Affichez toutes les adresses possibles

Exercice N°55. Adresse ip de classe C *

Soit une adresse réseau de classe C : 192.168.1.0.

- Affichez toutes les adresses ip (de 192.168.1.0 à 192.168.1.255). Attention au formatage de texte et utiliser soit les f strings soit la fonction `str` qui convertit un nombre en chaîne

Exercice N°56. Adresse ip de classe B **

Soit une adresse réseau de classe C : 172.168.1.0.

- Affichez toutes les adresses ip (de 192.168.1.0 à 192.168.1.255). Attention au formatage de texte et utiliser soit les f strings soit la fonction `str` qui convertit un nombre en chaîne

```
for x in range(256):  
    print("192.168.12."+str(x))  
  
for xx in range(256):  
    for x in range(256):  
        print("172.16."+str(xx)+"."+str(x))
```

V. LES TABLEAUX

Exercice N°57. Tableau : initialisation à zéro *

Il peut-être important de connaître les valeurs par défaut qui se trouvent dans un tableau. Arduino initialise toutes les variables à zéro, mais cela n'est pas vrai pour tous les compilateurs.

- Le programme initialise un tableau nommé `Tab` de 10 valeurs entières à 10 en utilisant une boucle `for`

Exercice N°58. Eléments > à 10 *

Soit un tableau d'entiers que vous remplierez au début du programme, écrire un programme qui va compter le nombre d'éléments supérieurs à 10.

Exercice N°59. Pairs et impairs **

Soit un tableau de 10 entiers (par exemple 1,3,4,6,3,2,0,...). Écrire un programme qui range les éléments de ce tableau dans deux autres tableaux, en mettant les éléments pairs à gauche et les éléments impairs à droite



Exercice N°60. Le plus grand **

En utilisant le programme précédent, trouver l'élément maximal de ce tableau.

Exercice N°61. Tableaux à 2 dimensions ***

1. Demander à l'utilisateur des valeurs du tableau en 2 dimensions (de 4 * 3 entiers) puis afficher toutes les valeurs

On utilisera dans cet exercice deux boucles imbriquées :

```
for( ..... ) {
    for( ..... ){
        action(s)
    }
}
```

Exercice N°62. Le plus grand/petit, la moyenne **

- Demander à l'utilisateur 10 notes et les mettre dans un tableau
- Parcourir le tableau et trouvez le plus grand élément
- Faire de même avec le plus petit
- Et pour finir calculer la moyenne

Exercice N°63. Répertoire téléphonique d'un hôtel ***

On désire mettre en place le répertoire téléphonique d'un hôtel : à chaque pièce correspond un numéro de téléphone

Nom	Nom abrégé	Numéro
Réception	Rec	9
Bureau rdc	BurRdc	1
Salle de conférence	SalleConf	2
Bureau N°1 1 ^{er} étage	Bureau11	11
Bureau N°2 1 ^{er} étage	Bureau21	12
Bureau N°1 2 ^{ème} étage	Bureau12	21
Bureau N°2 2 ^{ème} étage	Bureau22	22

Vous utiliserez trois tableaux : l'un contenant les noms, le deuxième contenant les noms abrégés et le dernier les numéros de téléphone, tels que le numéro tel[i] soit le numéro de téléphone de la pièce dont le nom est nom[i] (le nom abrégé nomCourt[i]).



Exercice N°64. Répertoire téléphonique d'un hôtel (niveau II) ***

Compléter le programme : demander le nom abrégé et afficher le nom complet de la pièce ainsi que le numéro de téléphone.
Utiliser la fonction strcmp.

Exercice N°65. Manipulation de tableaux ***

Soit un tableau à deux dimensions :

5	6	3	4
5	2	0	3
1	2	4	3
7	8	9	8

- Déclarer ce tableau au début du programme
- L'afficher
- Afficher uniquement toutes les lignes paires
- Afficher les éléments de la diagonales (de haut à gauche jusqu'en bas à droite)

Exercice N°66. Recherche d'un élément ***

Vous disposez d'un tableau de n éléments correspondant aux prénoms de vos amis et vous voulez savoir si l'un de ceux-ci est bien présent dans votre tableau. Il faut alors le rechercher. Le principe consiste à balayer l'intégralité du tableau à l'aide d'une structure itérative et à en sortir dès que l'élément a été trouvé ou que le nombre maximal d'indice a été dépassé. À la sortie de la boucle, il faudra de nouveau vérifier pour savoir si oui ou non l'élément a été trouvé : il se peut en effet que tout le tableau ait été parcouru et que ce soit la raison de la sortie de la boucle.

Exercice N°67. Tableau à deux dimensions ***

- créez deux tableaux à deux dimensions (nommés Tableau1 et Tableau2) de 5 entiers sur 3
- Remplissez les tous les deux
- Les afficher
- Affichez la plus grande valeur des deux tableaux

VI. LES FONCTIONS

a. Ressemblant aux exercices précédents

Exercice N°68. Afficher 5 fois *

Ecrire et utiliser une fonction *AfficherMot* qui va écrire :

Coucou N°1
Coucou N°2
Coucou N°3
Coucou N°4



Coucou N°5

Exercice N°69. Afficher n fois *

Même question que ci-dessus, mais passer en paramètre le nombre de fois que coucou sera affiché

Exercice N°70. Conversion ko en bits *

Écrivez une fonction qui demande un nombre en ko et renvoie la valeur correspondante en bits

b. Un peu de math

Exercice N°71. Factorielle *

Ecrire une fonction *Factorielle* qui prend en paramètre un entier n, et retourne la factorielle.

Exercice N°72. Table de multiplication *

Ecrivez la fonction TableMultiplication_7() dont l'affichage est :

```
>>> TableMultiplication_7 ()
0 *7= 0
1 *7= 7
2 *7= 14
3 *7= 21
4 *7= 28
5 *7= 35
6 *7= 42
7 *7= 49
8 *7= 56
9 *7= 63
10 *7= 70
```

Exercice N°73. Table de multiplication v2 *

Ecrivez la fonction TableMultiplication(multiplicateur) dont l'affichage est :

```
>>> TableMultiplication(8)
0 * 8 = 0
1 * 8 = 8
2 * 8 = 16
3 * 8 = 24
4 * 8 = 32
5 * 8 = 40
6 * 8 = 48
7 * 8 = 56
8 * 8 = 64
9 * 8 = 72
10 * 8 = 80
```



Exercice N°74. Est pair *

Écrire une fonction `entier_est_pair()` qui teste si un nombre entier naturel est pair. Cette fonction prendra en paramètre un entier naturel et renverra le booléen `True` (1) s'il est pair, `False` (0) sinon.

Exercice N°75. Est impair *

Même question mais pour un nombre impair.

Exercice N°76. Est premier

Écrire une fonction `entier_est_premier()` qui teste si un nombre entier naturel est premier, c'est-à-dire s'il n'est divisible que par 1 et par lui-même. Cette fonction prendra en paramètre un entier naturel et renverra le booléen `True` s'il est premier, `False` sinon.

AIDE : on pourra naturellement tester la divisibilité de l'entier n potentiellement premier par l'ensemble des entiers naturels qui lui sont inférieurs. Mais pour une meilleure efficacité, on pourra se convaincre qu'il n'y aura besoin que de tester la divisibilité de n par les entiers naturels k inférieurs ou égaux à \sqrt{n} .

Exercice N°77. Premiers suite

Écrire une fonction `premiers()` qui prend en paramètre un entier n et renvoie la liste des n premiers nombres premiers.

Écrire une fonction `premiers_inf()` qui prend en paramètre un entier n et renvoie la liste de tous les nombres premiers inférieurs ou égaux à n .

Exercice N°78. Liste chiffres

Écrire une fonction `liste_chiffres()` qui renvoie dans une liste chacun des chiffres qui composent le nombre Entier passé en paramètre.

Exercice N°79. Boucle inverse *

Implémenter une fonction `BoucleInverse` qui prend en paramètre un entier n , et affiche les entiers de $n-1$ à 0 sur le terminal.

Exercice N°80. DixNombres *

Écrire un programme qui demande un entier de départ, et qui ensuite affiche les dix entiers suivants.

Par exemple : en entrant 17, l'ordinateur affichera les entiers de 18 à 27.

Implémenter une fonction `DixNombres` réalisant cet algorithme.

Exercice N°81. La fonction MoyenneTroisNotes *

Cette fonction demande comme paramètres trois notes (réelles) et retourne leur moyenne

Exercice N°82. Afficher n fois un mot **

Maintenant le mot sera également passé en paramètre.

Exercice N°83. Occurrence **

Écrire une fonction occurrences() donnant le nombre de fois où un élément figure dans un tableau. Les paramètres de cette fonction seront, dans l'ordre, le tableau, une valeur. La valeur de retour sera un entier naturel (-1 si l'élément n'apparaît pas dans le tableau)

Exercice N°84. Conversion ***

Soit la fonction ConversionSecondes qui convertit un nombre d'heures, de minutes et de secondes en secondes.

Cette fonction demande donc ces trois paramètres et retourne le nombre total de secondes.

Exercice N°85. Fonction maximum

Définissez une fonction maximum(n1,n2,n3) qui renvoie le plus grand de 3 nombres n1, n2, n3 fournis en arguments. Par exemple, l'exécution de l'instruction :

```
printf("%d\n",maximum(22,55,25));
```

doit donner le résultat : 55.

Exercice N°86. Fonction : taille d'une image **

La taille d'une image peut être calculé ainsi : nombre de pixels en largeur * nombre de pixels en hauteur*taille d'un pixel

Par exemple



La taille est : $300 \times 200 \times 24 \text{ bits/pixels} = 1440000 \text{ bits} = 1440000/8 = 180000 \text{ octets} = 180000/1024 = 175.8 \text{ ko}$

🔧 Créez une fonction dont les paramètres sont : le nombre de pixels en largeur, en hauteur et le nombre de bits/pixels puis renvoie la taille en ko.

🔧 Testez cette fonction et affichez les résultats

```
def TailleImage(largeur, hauteur, bitsPixel):
    return largeur*hauteur*bitsPixel/(8*1024)
```

	BTS SN - EC	
	- Langage C -	
	Cahier d'exercices	

```
taille=TailleImage(300,200,24)
print(f"La taille de l'image est :{taille:6.2f}")
```

c. En lien avec les tableaux

Exercice N°87. Fonction moyenne **

Ecrire une fonction *moyenne* qui prend comme paramètre un tableau de nombre ainsi que sa taille et renvoie la moyenne.

Exercice N°88. Fonction minimum *

Écrire une fonction *minimum()* qui prend en paramètre une liste ou tableau de nombres et renvoie la valeur minimale de ces nombres

Exercice N°89. Fonction position_minimum **

Écrire une fonction *position_minimum()* qui prend en paramètre une liste PYTHON de nombres et renvoie la position de la valeur minimale de ces nombres.

Exercice N°90. Fonction contient_valeur *

Ecrire cette fonction qui prend comme en paramètre une liste de nom, sa taille et une valeur puis renvoie True (1) si la valeur est trouvé et False (0) sinon.

Exercice N°91. Fonction position_valeur *

Ecrire cette fonction qui prend comme en paramètre une liste de nom, sa taille et une valeur puis renvoie sa position ou -1 si la valeur n'est pas trouvée.

Exercice N°92. Médiane **

Écrire une fonction *mediane()* calculant la médiane d'un tableau de valeurs numériques. Le mode de Calcul de la médiane d'une série statistique peut notamment être retrouvé sur Wikipedia, exemples à l'appui:

[https://fr.wikipedia.org/wiki/Médiane_\(statistiques\)#Mode_de_calcul](https://fr.wikipedia.org/wiki/Médiane_(statistiques)#Mode_de_calcul)

Exercice N°93. Valeurs_sup ***

Écrire une fonction *valeurs_sup()* qui retourne les éléments d'une liste qui sont supérieurs ou égaux à une valeur donnée. Cette fonction prendra deux paramètres : d'un tableau et une valeur. Elle renverra un tableau, éventuellement vide.

Par exemple si le tableau est composé de 1,2,3 et la valeur 2.5, la fonction renvoie 3



Exercice N°94. Valeur_sup_moyenne **

Écrire une fonction `valeurs_sup_moy()` qui retourne les éléments d'une liste qui sont supérieurs ou égaux à la moyenne de la liste.

Exercice N°95. Renverse **

Écrire une fonction `renverse()` qui prend en argument une liste et retourne une liste constituée des éléments rangés dans l'ordre inverse.

Exercice N°96. Est_croissante **

Écrire une fonction `est_croissante()` qui prend en argument une liste et renvoie le booléen True (1) si cette liste est composée d'éléments rangés par ordre croissant, False (0) sinon.

Exercice N°97. Est_decroissante **

Écrire une fonction `est_decroissante()` qui prend en argument une liste et renvoie le booléen True (1) si cette liste est composée d'éléments rangés par ordre décroissant, False (0) sinon.

Exercice N°98. plus_longue_suite_croissante() ***

Écrire une fonction `plus_longue_suite_croissante()` qui retourne une liste comprenant la plus longue suite d'éléments successifs qui se suivent dans un ordre croissant. S'il existe plusieurs réponses possibles, la première suite trouvée aura la priorité.

Par exemple si la liste est 1, 5, 2, 3, 4, 0, 6, la fonction retourne 2,3,4

VII. LES CHAINES DE CARACTERES

a. En langage C

Une chaîne de caractères (string en anglais) est une suite de caractères stockée dans un tableau de char et terminée par le caractère `'\0'`.

Le `'\0'` marque la fin de la chaîne quelle que soit la taille du tableau mais le nombre total de caractères dans la chaîne ne peut pas dépasser la taille du tableau.

Par exemple :

1. `char Str1[15];`
2. `char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};`
3. `char Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'};`
4. `char Str4[] = "arduino";`
5. `char Str5[8] = "arduino";`
6. `char Str6[15] = "arduino";`



1. Str1 : déclare un tableau de char sans initialisation.
2. Str2 : déclare un tableau de char (avec un élément supplémentaire) et le compilateur va rajouter le caractère null obligatoire
3. Str3 : ajout explicite du caractère null
4. Str4 : initialise avec une chaîne de caractère entre guillemets (double), le compilateur va dimensionner le tableau pour l'ajuster à la taille de la chaîne et va ajouter le caractère null
5. Str5 : initialise le tableau avec une taille explicite et une chaîne de caractères.
6. Str6 : initialise le tableau et laisse un espace supplémentaire pour une chaîne de caractères plus importante.

Exercice N°99. Utilisation des fonctions de manipulations de chaînes de caractères ***

Utiliser les fonctions de manipulation de chaîne de caractères :

<https://openclassrooms.com/courses/apprenez-a-programmer-en-c/les-chaines-de-caracteres#/id/r-15764>

À partir d'un menu proposé à l'utilisateur, écrire le programme suivant :

- Saisir une chaîne de caractères.
- Compter la longueur d'une chaîne.
- Comparer 2 chaînes à saisir. Il s'agit de dire si elles sont identiques et laquelle est la première dans l'ordre lexicographique.
- Concaténer 2 chaînes dans la première (attention aux débordements).

b. Concernant les dates avec des fonctions

Exercice N°100. Fonction mois *

Définissez une fonction Mois(n) qui renvoie le nom du mois de l'année.

Par exemple :

nomMois(3) ; doit donner le résultat : Mars.

Exercice N°101. Heure_chaine **

Écrire une fonction ayant comme paramètre une heure, passée sous forme tableau et renvoyant (en Python) ou affichera (en C) une chaîne de caractères convenablement formatée représentant l'heure.

Par exemple si le tableau est 22,12,55 la chaîne sera "22 h 12 min 55 s"

Note : on utilisera sprintf en C ou str en Python.

Exercice N°102. Heure_chaine **

Écrire une fonction ayant comme paramètre une date, passée sous forme tableau et renvoyant une chaîne de caractères convenablement formatée représentant la date .

Par exemple si le tableau est 2022,8,22 la chaîne sera "22 août 2022".



Note : on utilisera sprintf en C ou str en Python.

Exercice N°103. AjouterAnnees **

Écrire une fonction ajouterAnnees () ayant comme paramètres une date sous la forme d'un tableau à trois valeurs (année, mois, jour), suivie d'un entier relatif (donc positif ou négatif) représentant un nombre d'années à ajouter (si ce nombre est négatif, cela reviendra à retrancher du temps). La fonction renverra un tableau représentant une nouvelle date

Par exemple si le tableau est : 2022,8,22 et le nombre : -2 le tableau renvoyé sera 2020, 8 , 22

Exercice N°104. AjouterMois **

Même exercice mais avec les mois

Exercice N°105. AjouterJour ***

Même exercice mais avec les jours

Exercice N°106. AjouterHeures ***

Écrire une fonction AjouterHeures ayant comme paramètres une date-heure sous la forme d'un tableau à six valeurs (année, mois, jour, heure, minute, seconde), suivie d'un entier relatif (donc positif ou négatif) représentant un nombre d'heures à ajouter (si ce nombre est négatif, cela reviendra à retrancher du temps). La fonction renverra un tableau représentant une nouvelle date-heure

Par exemple si le tableau est : 2022,8,22,23,15,23 et le nombre : +2 le tableau renvoyé sera 2022,8,23,1,15,23

Exercice N°107. AjouterMinutes ***

Même exercice mais avec les minutes

Exercice N°108. AjouterSecondes ***

Même exercice mais avec les secondes

VIII. LES STRUCTS

Exercice N°109. Structure eleve ***

Un élève a un nom, un prénom, une date de naissance, et aura dix notes dans l'année.
Proposez une structure en C et le programme qui l'utilise.

Exercice N°41 bis Structure Classe avec élèves.

Dans une école primaire, une classe a un niveau, un instituteur, et contient vingt élèves.



Donnez les structures des enregistrements classe.

Écrivez un programme qui va afficher pour chaque classe la liste des élèves et leurs dix notes associées.

Exercice N°110. HeuresMnsSec ***

Soit une structure nommée HeuresMnsSec qui contient une heure sous le format hh :mm :ss. Grâce à elle, vous pouvez très facilement trouver le temps écoulé entre une première heure et une seconde heure. Le principe est des plus simples :

- Vous saisissez deux horaires dans des enregistrements hms.
- Vous les convertissez en nombre de secondes.
- Vous faites la différence entre ces deux nombres.

IX. CHAMPION DES MATHS ****

Vous allez réaliser un programme permettant la révision de notions mathématiques niveau CM1 (<http://championmath.free.fr>). Les mini-jeux seront, quand même niveau CM1 difficile.

Exercice N°111. Page d'accueil

Reproduire cette page d'accueil :

```
+-----+
|1 : Addition          |
|2 : Soustraction     |
|3 : Multiplication    |
|4 : Tables des multiplications |
|5 : Divisions        |
|0 : Sortir du jeu     |
+-----+
```

Quel est votre choix ?

Exercice N°112. Choix multiples

Afficher "Addition" pour le choix 1, "Soustraction" pour le choix 2, etc...

Pour cette question, utiliser obligatoirement des if...then...else

Exercice N°113. Addition

Le joueur doit trouver l'addition de deux nombres entiers compris entre 0 et 100. Ce mini-jeu diffère de celui du site, car il ne propose que des entiers.

L'affichage sera :

Quel est votre choix ?



1

Addition

89+49 = ?

Entrez le résultat

Vous avez entré 138

Bravo

Aides :

- Afficher « Perdu » si l'utilisateur s'est trompé dans les calculs
- Mettre une temporisation de 3 secondes pour voir le résultat (`sleep(3);`)
- Afficher ("un nombre aléatoire entre 0 et 10")`srand (time(NULL)); // indispensable1`
- `printf("nombre aléatoire entre 0 et 10 : %d ", rand() % 11);`
- `printf("nombre aléatoire entre 1 et 10 : %d ", rand() % 10 + 1);`

Exercice N°114. multiplication

Idem que le mini-jeu addition, avec des nombres entiers compris entre 1 et 10.

Exercice N°115. Soustraction

Les nombres seront compris entre 0 et 100. Attention, le résultat doit être toujours positif, c'est-à-dire que le premier nombre doit être plus grand que le second (utiliser une permutation vue en cours et en exercice)

Exercice N°116. Sortie du jeu

Tant que le joueur n'a pas choisi 0 (Sortir du jeu), la page d'accueil est affichée. Dans le cas contraire, il est affiché "Merci de votre visite".

Exercice N°117. Utiliser un switch case

Remplacer les if then else de la phase N°1 par un seul switch case.

Exercice N°118. Gestion des points

A chaque fin de mini-jeux, le joueur reçoit un point en plus. Si le nombre de points est supérieur ou égal à 10, alors afficher "Vous avez gagné" puis arrêter le jeu.

Exercice N°119. Tables de multiplication

¹ En langage C, l'initialisation du générateur de nombres aléatoires, obligatoire sinon on aura toujours le même nombre aléatoire à chaque exécution du programme



Le joueur commence par choisir la table sur laquelle il veut travailler (table des 1 à 10) ensuite on affiche la table de multiplication.

Exercice N°120. Tables de multiplication (suite)

Après l'affichage de la table, le joueur pourra s'exercer en donnant, dans l'ordre croissant, le résultat de la table (dans notre exemple $2*1 = ?$ puis $2*2 = ?$ etc...). Afficher « Perdu » ou « Gagné »

Pour vos tests, n'oubliez pas de mettre un sleep à la fin.

X. AUTRES PROBLEMES NIVEAU ****

Exercice N°121. Le coffre-fort ****

On désire ouvrir un coffre-fort. Il faut une clef et un code. On a retrouvé une clef, mais pas le code. On sait seulement que la combinaison est formée de deux nombres compris entre 0 et 99 inclus (*Code1* et *Code2*) et que le coffre-fort s'ouvre :

- si le produit des deux nombres est 1638 (*condition1*)
- et si la somme est comprise entre 75 et 105 *condition2*)
- et si *Code1* est un multiple de 7 et *Code2*, un multiple de 13 (*condition3*)

Les trois conditions doivent être vérifiées pour que le coffre-fort s'ouvre.

- a. Ecrire en fonction de *Code1* et *Code2*, les expressions booléennes des trois conditions d'ouverture (*condition1*, *condition2*, *condition3*)
- b. Ecrire en fonction de *condition1*, *condition2*, *condition3* l'expression booléenne *Coffre_Ouvert* qui vaut vrai si le coffre-fort peut être ouvert faux sinon.
- c. Ecrire en fonction de *Code1* et *Code2*, l'expression booléenne *Intervalle_OK* qui vaut vrai si chaque nombre est compris entre [0,99] inclus, faux sinon.
- d. Ecrire un programme qui demande deux nombres, vérifie s'ils sont dans l'intervalle, et affiche « Coffre-fort ouvert » si *Coffre_Ouvert* est vrai, et « Coffre-fort toujours fermé » sinon.

Question Bonus : pour connaître toutes les combinaisons, il serait fastidieux d'essayer toutes les valeurs de *Code1* et *Code2* ! Aussi, écrivez un programme qui essaye toutes les combinaisons (utilisez deux boucles imbriquées).

Exercice N°122. Problème : parité d'un nombre binaire ****

Un nombre binaire est dit pair si le nombre de 1 qui le compose est pair ; elle est dites impair sinon. Aussi on ajoute un bit, nommé *p*, au nombre binaire tel que :

- $p = 0$ si le nombre de 1 est pair



- $p = 1$ sinon.

Ce bit est ajouté à droite du bit de poids faible.

Par exemple :

- 01010110 est pair car le nombre de 1 est 4 qui est un nombre pair. Aussi, $p=0$ et le nombre binaire s'écrit 010101100 (01010110 et 0)
- 11011001 est impair car le nombre de 1 est 5 qui est impair, $p=1$ et le nombre binaire s'écrit 110110011. (11011001 et 1)

Ainsi le nombre sous forme binaire est toujours pair ! (010000111 est composé de 4 '1' et 110110011 est composé de 6 '1'.

Afin de détecter certaines erreurs dans la transmission d'un nombre binaire, on peut appliquer la technique de contrôle de parité

- A l'émission, on ajoute le bit p au nombre binaire
- A la réception on vérifie la parité du nombre binaire et si elle est impaire alors il y a une erreur.

1 - Emission des données

Dans cette partie, l'émission du message va être simulée en appliquant le contrôle de parité.

a. Complétez le tableau suivant :

Nombre au format binaire	Parité bit p	Nombre+parité
01010111		
01011100		
00001110		
00110101		

- Ecrivez un programme qui demande à l'utilisateur 8 chiffres (entiers) puis les affiche. Il faut vérifier que $a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0$ soient des nombres binaires (0 ou 1)
- Affichez un message d'erreur dans le cas où un des nombres n'est pas binaire
- Calculez et affichez la parité (utilisez $\%2==0$)
- Affichez le nombre modifié au format binaire
- Affichez le nombre modifié au format décimal.
- Mettez ce code dans une fonction et écrivez le programme qui l'appelle.

2 – Réception des données

Maintenant on désire vérifier la parité du nombre et donc l'intégrité du message.

- Ecrire une fonction qui demande en entrée les bits (entiers) $a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0$ et p et affiche « Bonne transmission » si la parité est vérifiée et « Erreurs détectées » sinon. (La saisie des nombres ne se fait pas dans cette fonction.)
- Améliorez l'algorithme précédent en demandant le nombre sous la forme décimale (compris entre 0 et 511) et affiche « Bonne transmission » si la parité est vérifiée et « Erreurs détectées » sinon.



Exercice N°123. Assurance ****

Une compagnie d'assurance automobile propose à ses clients quatre familles de tarif identifiables par une couleur ; du moins au plus onéreux.

- tarif vert
- tarif orange
- tarif rouge

La famille de tarif dépend de la situation du conducteur :

- Un conducteur de moins de 25 ans se voit attribuer le tarif rouge si toutefois il n'a jamais été responsable d'accident ; sinon la compagnie refuse de l'assurer.
- Un conducteur de plus de 25 ans mais titulaire depuis moins de 2 ans, a le droit au tarif orange, s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident ; sinon il est refusé.
- Un conducteur de plus de 25 ans, titulaire du permis depuis plus de 2 ans, bénéficie du tarif vert s'il n'est pas à l'origine d'aucun accident, du tarif orange pour un accident, du tarif rouge pour 2 accidents, et refusé au-delà.

Ecrivez un programme qui demande les renseignements nécessaires à l'utilisateur et qui affiche la couleur du tarif ou le refus de l'assurance.