**Travaux pratiques 4 programmation en langage C- Les tableaux et les chaines de caractères**

## Exercice 1

Déclarer un tableau d'entiers de 10 éléments et l'initialiser avec les nombres 1 à 10.

Afficher le tableau en séparant les valeurs par des virgules.

*Corrigé.*

## *#include <stdio.h> // Inclure la bibliothèque STanDard Input Output*

## *// permet d'utiliser, entre autre, printf et scanf*

## *int main() {*

## *int tab[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};*

## 

## *printf("%d", tab[0]);*

## *for(int i = 1 ; i < 10 ; i++) printf(", %d", tab[i]);*

## *return 0;*

## *}*

## Exercice 2

Déclarer un tableau d'entiers de 100 éléments et l'initialiser avec les nombres 0 à 99 (utiliser une boucle !).

Afficher le tableau en séparant les valeurs par des virgules (limiter à 10 valeurs par lignes).

Résultat attendu :

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

10, 11, 12, 13, 14, 15, 16, 17, 18, 19

20, 21, 22, 23, 24, 25, 26, 27, 28, 29

30, 31, 32, 33, 34, 35, 36, 37, 38, 39

40, 41, 42, 43, 44, 45, 46, 47, 48, 49

50, 51, 52, 53, 54, 55, 56, 57, 58, 59

60, 61, 62, 63, 64, 65, 66, 67, 68, 69

70, 71, 72, 73, 74, 75, 76, 77, 78, 79

80, 81, 82, 83, 84, 85, 86, 87, 88, 89

90, 91, 92, 93, 94, 95, 96, 97, 98, 99

*Corrigé.*

*#include <stdio.h> // Inclure la bibliothèque STanDard Input Output*

*// permet d'utiliser, entre autre, printf et scanf*

*int main() {*

*int tab[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};*

*printf("%d", tab[0]);*

*for(int i = 1 ; i < 10 ; i++) printf(", %d", tab[i]);*

*return 0;*

*}*

## Exercice 3

Soient deux tableaux de nombres réels **tSource** et **tDestination** de 10 éléments chacun.

Écrire un programme permettant de recopier, dans **tDestination**, tous les éléments positifs de **tSource**, en complétant éventuellement **tDestination** par des zéros (initialiser **tSource** avec des valeurs au moment de sa déclaration).

Afficher les deux tableaux.

*Corrigé.*

*#include <stdio.h> // Inclure la bibliothèque STanDard Input Output*

*// permet d'utiliser, entre autre, printf et scanf*

*int main() {*

*const int TAB\_SIZE = 10;*

*float tDestination[TAB\_SIZE];*

*float tSource[] = {12.5, -25.4, -78.9, 45.1, 8.6, -1.0, 37.2, -51.3, 88.8, 94.7};*

*int iSrc, iDst;*

*// Recopier les éléments positifs de tSource dans tDestination*

*for(iSrc = 0, iDst = 0 ; iSrc < TAB\_SIZE ; iSrc++) {*

*if(tSource[iSrc] > 0) tDestination[iDst++] = tSource[iSrc];*

*}*

*// Compléter tDestination avec des zéros*

*while(iDst < TAB\_SIZE) tDestination[iDst++] = 0;*

*// Aficher les tableaux*

*printf("tSource : %.2f", tSource[0]);*

*for(int i = 1 ; i < TAB\_SIZE ; i++) printf(", %.2f", tSource[i]);*

*printf("\ntDestination : %.2f", tDestination[0]);*

*for(int i = 1 ; i < TAB\_SIZE ; i++) printf(", %.2f", tDestination[i]);*

*return 0;*

*}*

## Exercice 4

Écrire un programme qui demande 10 nombres entiers à l’utilisateur, les range dans un tableau avant d’en rechercher le plus grand et le plus petit.

Afficher le tableau, ainsi que le nombre le plus petit et le plus grand.

*Corrigé.*

## #include <stdio.h> // Inclure la bibliothèque STanDard Input Output

## // permet d'utiliser, entre autre, printf et scanf

## int main() {

## const int TAB\_SIZE = 10;

## int tNombres[TAB\_SIZE];

## 

## // Demander les nombres à l'utilisateur

## printf("Entrez les nombres (entiers) du tableau\n");

## for(int i = 0 ; i < TAB\_SIZE ; i++) {

## printf("Nombre %d/10 : ", i+1);

## scanf("%d", &tNombres[i]);

## }

## 

## // Rechercher le plus petit nombre

## int nbMin = tNombres[0];

## for(int i = 1 ; i < TAB\_SIZE ; i++) {

## if(tNombres[i] < nbMin) nbMin = tNombres[i];

## }

## // Rechercher le plus grand nombre

## int nbMax = tNombres[0];

## for(int i = 1 ; i < TAB\_SIZE ; i++) {

## if(tNombres[i] > nbMax) nbMax = tNombres[i];

## }

## 

## // Aficher le tableau

## printf("\ntNombres : %d", tNombres[0]);

## for(int i = 1 ; i < TAB\_SIZE ; i++) printf(", %d", tNombres[i]);

## 

## printf("\nLa valeur minimale est : %d", nbMin);

## printf("\nLa valeur maximale est : %d", nbMax);

## return 0;

## }

**Exercice 5**

Demander à l’utilisateur de saisir des notes (entre 0 et 20) et lui expliquer qu’une valeur hors de cet intervalle arrêtera la saisie.

1. Saisir les notes et les mémoriser dans un tableau
2. Compter les notes saisies et afficher leur nombre
3. Calculer et afficher la moyenne
4. Comparer chaque note à la moyenne et ajouter, dans l’affichage précédent "égal", "inférieur" ou "supérieur à la moyenne"
5. Compter et afficher combien il y a de notes supérieures à la moyenne
6. Dans le tableau de notes, chercher la note la plus petite. Afficher cette note et sa position dans le tableau
7. Même chose pour la note la plus grande.

Le programme affichera un message d'erreur si le nombre de note saisi est 0.

*Corrigé.*

## *#include <stdio.h> // Inclure la bibliothèque STanDard Input Output*

## *// permet d'utiliser, entre autre, printf et scanf*

## *int main() {*

## *const int NB\_MAX\_NOTES = 50;*

## *int nbNotes;*

## *float note, sommeDesNotes;*

## *float tabNotes[NB\_MAX\_NOTES]; // Déclare un tableau de 50 réels*

## 

## *printf("Entrez des notes (entre 0 et 20)\nPour stopper, saisir une note hors de cet interval.\n");*

## 

## *nbNotes = 0;*

## *// A. Saisir les notes et les mémoriser dans un tableau*

## *// + B. Compter les notes saisies et afficher leur nombre*

## *do {*

## *scanf("%f", &note);*

## *tabNotes[nbNotes] = note; // tabNotes[nbNotes] <-- note*

## *nbNotes++;*

## *} while((note >= 0) && (note <= 20) && (nbNotes < NB\_MAX\_NOTES));*

## 

## *if((note < 0) || (note > 20)) nbNotes--; // La dernière note saisie n'est pas valide*

## *else printf("\nVous avez ateint le nombre maximum de notes possible.");*

## *printf("\nVous avez saisi %d notes", nbNotes);*

## 

## *// C. Calcul et affichage de la moyenne*

## *sommeDesNotes = 0;*

## *int i;*

## *for(i = 0 ; i < nbNotes ; i++) {*

## *sommeDesNotes = sommeDesNotes + tabNotes[i];*

## *}*

## 

## *float moyenne = sommeDesNotes / nbNotes;*

## *printf("\n\nLa moyenne des notes est : %.2f\n", moyenne);*

## 

## *// D. Comparer chaque note à la moyenne et ajouter, dans l’affichage précédent "égal",*

## *// "inférieur" ou "supérieur à la moyenne"*

## *// + E. Compter et afficher combien il y a de notes supérieures à la moyenne*

## *int nbNotesSupMoy = 0;*

## *for(i = 0 ; i < nbNotes ; i++) {*

## *printf("\n%d : %.2f", i+1, tabNotes[i]);*

## 

## *if(tabNotes[i] > moyenne) {*

## *printf(" > à la moyenne");*

## *nbNotesSupMoy++;*

## *}*

## *else if(tabNotes[i] < moyenne)*

## *printf(" < à la moyenne");*

## *else*

## *printf(" = à la moyenne");*

## *}*

## 

## *printf("\n\nIl y a %d notes > à la moyenne.", nbNotesSupMoy);*

## 

## *// F. Dans le tableau de notes, chercher la note la plus petite*

## *float noteLaPlusFaible = 21;*

## *int positionNoteLaPlusFaible;*

## *for(i = 0 ; i < nbNotes ; i++) {*

## *if(tabNotes[i] < noteLaPlusFaible) {*

## *noteLaPlusFaible = tabNotes[i];*

## *positionNoteLaPlusFaible = i + 1;*

## *}*

## *}*

## 

## *// F. (suite) Afficher cette note et sa position dans le tableau*

## *printf("\n\nLa note la plus basse est %.2f, sa position est %d", noteLaPlusFaible, positionNoteLaPlusFaible);*

## *// G. Dans le tableau de notes, chercher la note la plus haute*

## *float noteLaPlusHaute = -1;*

## *int positionNoteLaPlusHaute;*

## *for(i = 0 ; i < nbNotes ; i++) {*

## *if(tabNotes[i] > noteLaPlusHaute) {*

## *noteLaPlusHaute = tabNotes[i];*

## *positionNoteLaPlusHaute = i + 1;*

## *}*

## *}*

## 

## *// G. (suite) Afficher cette note et sa position dans le tableau*

## *printf("\n\nLa note la plus haute est %.2f, sa position est %d", noteLaPlusHaute, positionNoteLaPlusHaute);*

## 

## *return 0;*

## *}*

**Exercice 6**

Nombre de caractères dans une chaîne

1. Demander à l’utilisateur de saisir son nom
2. Compter les caractères et afficher leur nombre

**Rappel :** un tableau de caractère (chaine de caractères), se termine par le caractère spécial : **'\0'**.

Corrigé.

#include <stdio.h> // Inclure la bibliothèque STanDard Input Output

// permet d'utiliser, entre autre, printf et scanf

int main() {

int nbCar;

char tNom[30];

printf("Saisissez votre nom : "); // Afficher : "Saisissez votre nom : "

scanf("%s", tNom); // Entrer : tNom[]

nbCar = 0; // nbCar <-- 0

while (tNom[nbCar] != '\0') { // Tant que(tNom[nbCar] != '\0')

nbCar++; // nbCar <-- nbCar + 1

} // Fin Tant que

// Afficher : "Votre nom compte nbCar caracteres"

printf("\nVotre nom compte %d caracteres", nbCar);

return 0;

}

## Exercice 7

Saisir un mot et l’afficher dans l’ordre inversé.

*Corrigé.*

#include <stdio.h> // Inclure la bibliothèque STanDard Input Output

// permet d'utiliser, entre autre, printf et scanf

int main()

{

// Variables :

int nbCar, i; // Entier : nbCar, i

char tMot[30]; // Tableau de caractères : tMot[30]

printf("Saisissez un mot : "); // Afficher : "Saisissez un mot : "

scanf("%s", tMot); // Entrer : tMot[]

// Déterminer la longueur du mot

nbCar = 0; // nbCar <-- 0

while(tMot[nbCar] != '\0') { // Tant que(tMot[nbCar] != '\0')

nbCar = nbCar + 1; // nbCar <-- nbCar + 1

} // Fin Tant que

// Affiche le mot caractère par caractère en partant de la fin

for(i = nbCar -1 ; i >= 0 ; i--) { // Pour i de nbCar-1 à 0 incrément -1

printf("%c", tMot[i]); // Afficher : "tMot[i]"

} // Fin Pour

return 0;

} // Fin

**Exercice 8**

Indice de masse corporelle

On imagine une visite médicale en deux parties où les patients se présentent d’abord tous pour la mesure de leur taille, puis repassent, dans le même ordre, au pesage.

1. Enregistrer dans un tableau la taille t en mètres de tous les patients qui se présentent (arrêt par la saisie d’un nombre <= 0).
2. Après la saisie, afficher le nombre total de patients.
3. Pour chaque patient précédemment mesuré, enregistrer la masse m en kilogrammes dans un second tableau.
4. Calculer et afficher la taille moyenne et le poids moyen des patients.
5. Pour chaque patient, calculer l’indice de masse corporelle : IMC = m/t² et afficher :
   * "pas assez" si IMC < 18.5,
   * "trop" si IMC > 25,
   * ou "normal" sinon.

*Corrigé.*

#include <stdio.h> // Inclure la bibliothèque STanDard Input Output

// permet d'utiliser, entre autre, printf et scanf

int main()

{

const int TAB\_SIZE = 30;

int i, nbPatient;

float tTaille[TAB\_SIZE], tMasse[TAB\_SIZE], imc, somme;

// A. Enregistrer la taille t en mètres de tous les patients qui se présentent

// (arrêt par la saisie d’un nombre ≤ 0)

printf("Entrez la taille des patients (en mètre) :\n");

i = 0;

do {

printf(" Taille du patient %d : ", i + 1);

scanf("%f", &tTaille[i]);

i = i + 1;

} while ((tTaille[i-1] > 0) && (i < TAB\_SIZE));

// B. Après la saisie, afficher le nombre total des patients.

nbPatient = i;

if(tTaille[i-1] < 0) nbPatient--; // La dernière taille saisie n'est pas valide

else printf("\nVous avez ateint le nombre maximum de %d patients possible.", TAB\_SIZE);

printf("\nIl y a %d patients.\n\n", nbPatient);

// C. Pour chaque patient précédemment mesuré, enregistrer la masse m en kilogrammes.

printf("Entrez la masse des patients (en kg) :\n");

for(i = 0; i < nbPatient; i++) {

printf(" Masse du patient %d : ", i + 1);

scanf("%f", &tMasse[i]);

}

// D. Calculer et afficher la taille et le poids moyens.

printf("\n");

// Taille moyenne

somme = 0;

for(i = 0 ; i < nbPatient ; i++) {

somme = somme + tTaille[i];

}

printf("\nLa taille moyenne des patients est : %.2f\n", somme / nbPatient);

// Poids moyen

somme = 0;

for(i = 0; i < nbPatient; i++) {

somme = somme + tMasse[i];

}

printf("\nLa masse moyenne des patients est : %.2f\n", somme / nbPatient);

// E. Pour chaque patient, calculer l’indice de masse corporelle : IMC = m/t\*t

// et afficher : pas assez , trop ou normal selon son IMC

for(i = 0; i < nbPatient; i++) {

imc = tMasse[i] / (tTaille[i] \* tTaille[i]);

printf("\nIMC patient %d : %.2f", i + 1, imc);

if(imc < 18.5) {

printf(" => Pas assez");

} else {

if(imc > 25) {

printf(" => Trop");

} else {

printf(" => Normal");

}

}

}

return 0;

}

**Exercice 9**

Vous devez réaliser la transmission d'une chaîne de caractères sous forme d'une trame selon des règles et un format précis.

La chaîne de caractère peut contenir uniquement les caractères suivants :

* Chiffres de '0' à '9'
* Lettres minuscules 'a' à 'z'
* Lettres majuscules 'A' à 'Z'

En cas de non respect de cette règle, l'utilisateur sera invité à recommencer la saisie.

En plus de la chaîne saisie par l'utilisateur, il faudra transmettre une somme de contrôle (checksum) qui permettra au récepteur de vérifier l'intégrité de la trame reçue.

Le checksum est la somme binaire, modulo 65536, de tous les caractères saisis. Le checksum est codé sur 2 octets selon le format big endian (octet de poids fort en premier). Il sera placé à la fin de la trame.

Le premier octet sera le caractère spécial STX (Start of TeXte), valeur 0x02.

Il sera suivi du nombre de caractères saisis par l'utilisateur sur 2 octets au format big endian

Le dernier octet sera le caractère spécial ETX (End of TeXte), valeur 0x03.

Réaliser le programme qui :

1. Permet à l'utilisateur de saisir la chaîne de caractères
2. Calcul le checksum
3. Affiche la trame à transmettre en hexadécimal (Exemple si l'utilisateur à saisie "Bonjour") :

Trame à tranmettre : 0x02 0x00 0x07 0x42 0x6F 0x6E 0x6A 0x6F 0x75 0x72 0x02 0xDF 0x03

**Remarque :** Pour afficher une valeur en hexadécimal avec **printf**, utiliser le format **"%X"**

*Corrigé*.

#include <stdio.h> // Inclure la bibliothèque STanDard Input Output

// permet d'utiliser, entre autre, printf et scanf

int main()

{

char trame[1000];

int saisieOk = 0;

int i, nbCar;

// A. Permet à l'utilisateur de saisir la chaîne de caractères

printf("Entrer une chaîne de caractères (caractères autorisés : 0-9A-Za-z) : ");

scanf("%s", trame);

do {

i = 0;

while(trame[i] != '\0') {

if(trame[i] < '0' || (trame[i] > '9' && trame[i] < 'A') || (trame[i] > 'Z' && trame[i] < 'a') || trame[i] > 'z') {

saisieOk = 0;

printf("\nErreur : la chaîne de caractères peut contenir uniquement les caractères 0-9A-Za-z.");

printf("\nSaisissez à nouveau : ");

scanf("%s", trame);

}

else {

saisieOk = 1;

}

i++;

}

} while(!saisieOk);

nbCar = i;

printf("saisie Ok (%d octets) : \n\t%s\n", nbCar, trame);

// B. Calcul le checksum

unsigned int checksum = 0;

for(i = 0 ; trame[i] != '\0' ; i++) checksum = (checksum + trame[i]) % 65536;

printf("Valeur du checksum : \n\t%d\n", checksum);

// C. Affiche la trame à transmettre en hexadécimal

printf("\nTrame à tranmettre : ");

printf("0x%02X ", 0x02); // STX

printf("0x%02X ", nbCar >> 8); // Nombre de caractères saisis (poids fort)

printf("0x%02X ", nbCar & 0xFF); // Nombre de caractères saisis (poids faible)

for(i = 0 ; trame[i] != '\0' ; i++) printf("0x%2X ", trame[i]);

printf("0x%02X ", checksum >> 8); // checksum (poids fort)

printf("0x%02X ", checksum & 0xFF); // checksum (poids faible)

printf("0x%02X", 0x03); // ETX

return 0;

}