Institut Notre-Dame Beauraing

UAA11

Le langage C

Livret d'exercices

Technique de transition

–

Option Informatique

–

Titulaire du cours : L. Embrechts

1. Exercices pratiques, premières instructions

1.1. Déclare, pour chaque type primitif[1], une variable contenant une valeur de ton choix. Ensuite, imprime à l'écran la valeur de chacune des variables.

1.2. Corrige chaque déclaration lorsque c'est nécessaire :

```c
int a = 3.14;
float b = 42;
char c = "A";
double d = 5.5;
char e = 'Z';
float f = 3.14;
double g = 10;
char h = 123;
int i = 2147483648;
float j = 5.67.8;
```

1.3. Soient les déclarations suivantes :

```c
int a = 10;
float b = 5.565;
char c = 'A';
```

Corrige les lignes suivantes lorsque c'est nécessaire :

```c
printf("Valeur de a: %d\n", a);
printf("Valeur de b: %d\n", b);
printf("Valeur de c: %c\n", c);
printf("Valeur de d: %f\n", d);
printf("Somme a+b: %c\n", a + b);
printf("Valeur absolue de -a: %d\n", a*-1);
printf("Valeur de c : %f\n", c);
printf("Produit a*d: %f\n", a * d);
printf("Valeur de b: %.2f\n", b);
printf("Valeur de b: %c\n", b);
```

1.4. Écris un programme qui demande deux entiers à l'utilisateur et qui affiche leur produit, leur somme, leur division ainsi que leur soustraction. *Pour quelle(s) valeur(s) le programme risque-t-il de renvoyer une erreur ?* Modifiez ensuite le programme afin de saisir deux réels.

1.5. Écris un programme qui échange deux entiers saisis. Ecris à l'écran avant et après l'échange :

`la variable X vaut : … la variable Y vaut : …`

1 Rappel : Les types primitifs en C sont : l'entier (`int`), le booléen (`bool`), le caractère (`char`) et le réel (`float` ou `double`)

1.6. Écris un programme qui demande à l'utilisateur de rentrer un caractère du clavier et qui affiche le code ASCII et hexadécimal de ce caractère de la manière suivante

```
caractère A          code ASCII = 65       hexadecimal = 41
```

1.7. Écris un programme qui détermine si un entier saisi est pair ou impair

1.8. Écris un programme qui demande trois caractères à l'utilisateur puis qui informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique. (On ne demande pas de restituer l'ordre alphabétique des trois caractères)

1.9. Écris un programme qui demande un nombre à l'utilisateur et l'informe ensuite si ce nombre est positif, négatif ou égal à zéro.

1.10. Ecris un programme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif, positif ou égal à zéro. Attention, à nouveau, on ne doit pas calculer le produit.

1.11. Écris un programme qui détermine le plus grand des trois entiers saisi.

1.12. Écris un programme qui affiche le plus grand et le plus petit d'une suite d'entiers saisi.
La suite de saisie au clavier se termine par l'encodage du zéro.
Astuce : Les nombres saisis ne sont pas tous conservés en mémoire.

2. Les structures conditionnelles

2.1. Écris un programme qui demande le jour de la semaine (saisie d'un nombre représentant le jour : lundi = 1 , mardi = 2, mercredi = 3…) et la température.
S'il fait plus de 14° degrés mais moins de 25, le programme répondra « il fait bon aujourd'hui! », s'il fait plus chaud, le programme répondra « Il fait chaud ajd ! » sinon qu'il fait froid. Si nous sommes un jour de semaine, le programme souhaitera à l'utilisateur une bonne semaine ou sinon un bon weekend. Si le programme ne reconnaît pas l'encodage du jour, le programme indiquera « Je n'ai pas compris quel jour nous sommes, mais bonne journée ». Exemple de dialogue :

```
Quel jour de la semaine sommes-nous ? (LUNDI=1, MARDI=2, MERCREDI=3,
JEUDI=4, VENDREDI=5, SAMEDI=6, DIMANCHE=7)
> 1
Quelle est la température ?
> 23
Bonjour,
Il fait bon aujourd'hui. Bonne semaine !
```

2.2. Ecris un programme qui demande à l'utilisateur le jour de la semaine (saisie d'un nombre représentant le jour : lundi = 1 , mardi = 2, mercredi = 3…). Le programme restitue ensuite le planning du jour.
Pour ce faire, utilisez la structure alternative switch.

Pourquoi la structure SWITCH est-elle la plus adaptée dans ce contexte ?

Lundi	test Nlds – 18h : Rdv avec Julie
Mardi	Interrogation de math
Mercredi	14h30 : VTT avec Ben
Jeudi	Redaction de Francais a finir
Vendredi	Soiree Beauraing
Samedi	Repas Nadine
Dimanche	Repos

2.3. L'école organise un souper et propose un menu enfant (moins de 12ans) à 8€50 et un menu adulte à 14€. Écris un programme qui demande à l'utilisateur le nombre d'enfants qui seront présents ainsi que le nombre d'adultes. Le dialogue se présentera de la manière suivante :

```
Bonjour, merci de participer a notre souper.

Combien d'adultes prendront part au souper ?

> 1

Combien d'enfants prendront part au souper ?

> 1
```

Le programme affichera alors le prix à payer à la famille sous la forme suivante. Veillez à afficher une réponse en nombre flottants à deux décimales de précisions.

```
Prix a payer : 22.50 euro
```

2.4. Un magasin de reprographie facture 0,10€ les dix premières photocopies, 0,09€ les vingts suivantes et 0,08€ au-delà. Écrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

2.5. Les habitants de Zorglub paient l'impôt selon les règles suivantes :

- Les hommes de plus de 20 ans paient l'impôt
- Les femmes paient l'impôt si elles ont entre 18 et 35 ans
- Les autres ne paient pas l'impôt

Le programme demandera donc l'âge et le sexe du Zorglubien (H/F) et se prononcera donc ensuite sur le fait que l'habitant est imposable ou non.

2.6. Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux : tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur :

- un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux 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 moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux 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 deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà
- de plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus de cinq ans. Ainsi, s'il satisfait à cette exigence, un client normalement "vert" devient "bleu", un client normalement "orange" devient "vert", et le "rouge" devient orange.

Écrire le programme permettant de saisir les données nécessaires (sans vérification de saisie) et de traiter ce problème.

3. Les structures itératives

3.1. Écrire un algorithme qui demande à l'utilisateur un nombre compris entre 20 et 25 jusqu'à ce que la réponse convienne.

3.2. Écrire un programme qui permette de saisir un nombre entier positif. Et ensuite, d'afficher la suite 1,2,3,4,...,n

3.3. Écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

3.4. Réécrire l'algorithme précédent, en utilisant cette fois l'instruction FOR/WHILE (celle que vous n'aurez pas utilisé).

3.5. Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

```
1 + 2 + 3 + 4 + 5 = 15
```

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

3.6. Créer un programme qui permette de calculer la factorielle d'un nombre saisi au clavier.
Quelle valeur maximale peut-être calculée ?

3.7. Créez un programme qui demande à l'utilisateur de saisir une série des nombres jusqu'à N entiers et qui affiche leur somme, leur produit et leur moyenne. Le nombre N est également à entrer au clavier.

3.8. Ecrire un programme qui permette de calculer la somme, le produit et la moyenne d'une suite de nombres positifs non nuls entrés au clavier, sachant que la suite est terminée par zéro. Bonus : si le nombre est négatif, indiquez à l'utilisateur un code d'erreur et proposez lui d'insérer un nouveau nombre.

3.9. La suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence par les termes 0 et 1 et ses premiers termes sont :

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	...	F_n
0	1	1	2	3	5	8	13		$F_{n-1}+F_{n-2}$

Crée un programme qui calcule la valeur du $n^{ième}$ terme de la suite. Cette fonction sera codée de manière itérative.

3.10. Créez un programme qui affiche les tables de multiplications d'un chiffre. Utilisez une structure itérative pour résoudre l'énoncé. Exemple du dialogue :

```
Entrez un chiffre :
> 3
La table de multiplication pour 3 :
    1 * 3 = 3
    2 * 3 = 6
    3 * 3 = 9
    …
    10 * 3 = 30
```

3.11. Vous organisez une soirée. Les tickets boissons se présentent sous deux catégories : les tickets bleus à 1,80 euros et les tickets rouges à 2,20 euros. Créez un logiciel qui affiche la liste des conversion de 1 à 40 tickets.
Utilisez une structure itérative pour résoudre l'énoncé. Attention, portez une attention particulière à l'orthographe de votre tableau. Exemple du dialogue :

```
1 ticket bleu: 1,80          |     1 ticket rouge : 2,20
2 tickets bleu: 3,60         |     2 tickets rouge : 4,40
3 tickets bleu: 5,40         |     3 tickets rouge : 6,60
4 tickets bleu: 7,20         |     4 tickets rouge : 8,80
5 tickets bleu: 9,00         |     5 tickets rouge : 10,10
…
```

3.12. Calcul d'une moyenne

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.

3.13. Donne la condition du while permettant de vérifier l'entrée de l'utilisateur sur base de la problématique donnée

Problématique	Condition du while
Demander à l'utilisateur d'entrer un chiffre entre 1 et 5.	
Demander à l'utilisateur d'entrer une lettre majuscule 'O' pour Oui ou 'N' pour Non.	
Demander à l'utilisateur d'entrer un nombre entre 10 et 20 inclus.	
Demander à l'utilisateur de choisir une option parmi 'A', 'B', 'C' ou 'D'.	
Demander à l'utilisateur d'entrer un nombre qui soit pair ou supérieur à 100.	
Demander à l'utilisateur d'entrer un caractère qui soit une lettre (majuscule ou minuscule).	
Demander à l'utilisateur d'entrer une température en Celsius qui soit soit négative, soit supérieure à 100.	
Demander à l'utilisateur d'entrer un nombre qui soit divisible par 3 ou par 5.	
Demander à l'utilisateur d'entrer un jour de la semaine (1-7).	
Demander à l'utilisateur d'entrer un caractère qui soit une voyelle (a, e, i, o, u, y).	
Demander à l'utilisateur d'entrer un code postal valide (entre 1000 et 9999 ou égal à 10000).	

3.14. Pour chaque scénario, complétez les deux dernières colonnes en indiquant les conditions de continuation et d'arrêt correspondantes.

Problématique et propositions	Je m'arrête si...	Je continue tant que...
Recherche d'une personne dans une liste de contacts A = J'ai trouvé la personne B = J'ai atteint la fin de la liste		
Attente d'un événement important A = Il est l'heure prévue B = Un message d'urgence est reçu		
Surveillance d'un système informatique A = Le système est en panne B = Une maintenance est programmée		
Processus de validation d'un document A = Toutes les signatures sont présentes B = Le délai est expiré		
Contrôle de qualité d'un produit A = Le produit est conforme aux normes B = Des défauts mineurs sont acceptables		
Processus d'admission à une université A = La moyenne est supérieure à 14 B = L'entretien est réussi		
Diagnostic médical A = Le patient a de la fièvre B = Le patient présente des symptômes respiratoires		
Processus de recrutement A = Le candidat a les compétences techniques B = Le candidat a une bonne attitude		
Système d'alarme A = Une intrusion est détectée B = Le système est en mode test		

3.15. En utilisant la structure itérative, imagine un système de résolution pour détecter si un chiffre est premier ou non. Expliquez en français, comment il serait possible de résoudre le problème.

- Commencez par définir un nombre premier.

- Ensuite, que doit-on tester sur ce nombre pour vérifier s'il est premier ?

Dans le programme, l'utilisateur insère un nombre. Le programme teste s'il est premier ou pas. Le programme écrit « le nombre inséré est premier » ou « le nombre inséré n'est pas premier ». Imagine un pseudo code qui résolve ce problème puis implémente ce code en C.

3.16. étoiles

3) Afficher un triangle rempli d'étoiles, s'étendant sur un nombre de lignes fourni en donnée et se présentant comme dans cet exemple :

```
*
**
***
****
*****
```

3.17. Écrire un programme qui affiche les tables de multiplication des nombres de 1 à 10, sous la forme suivante :

```
      |     1     2     3     4     5     6     7     8     9     10
-------------------------------------------------------------------
1     |     1     2     3     4     5     6     7     8     9     10
2     |     2     4     6     8     10    12    14    16    18    20
3     |     3     6     9     …
```

4. Les chaînes de caractères

4.1. Lesquelles des chaînes suivantes sont initialisées correctement ? Corrige les déclarations fausses et indiquez pour chaque chaîne de caractères le nombre **d'octets** qui sera réservé en mémoire.

- `char a[]   = "un\ndeux\ntrois\n";`
- `char b[12] = "un deux trois";`
- `char c[]   = 'abcdefg';`
- `char d[10] = 'x';`
- `char e[5]  = "cinq";`
- `char g[2]  = {'a', '\0'};`
- `char h[4]  = {'a', 'b', 'c'};`
- `char i[4]  = "'o'";`

4.2. Écris un programme qui demande le nom d'un utilisateur, le jour de la semaine et la température. Le programme souhaite la bienvenue à l'utilisateur. Ensuite, s'il fait plus de 14° degrés mais moins de 25, le programme répondra « il fait bon aujourd'hui! », s'il fait plus chaud, le programme répondra « Il fait chaud ajd ! » sinon qu'il fait froid. Si nous sommes un jour de semaine, le programme souhaitera à l'utilisateur une bonne semaine ou sinon un bon weekend. A noter, les jours de la semaines seront insérés de plein texte en majuscules, si le programme ne reconnaît pas l'encodage du nom de la semaine, le programme indiquera « Je n'ai pas compris quel jour nous sommes, mais bonne journée ». Exemple de dialogue :

```
Quel est ton prénom ?
> Tom
Quel jour de la semaine sommes-nous ? (LUNDI, MARDI, MERCREDI, JEUDI,
VENDREDI, SAMEDI, DIMANCHE)
> MARDI
Quelle est la température ?
> 23
Bonjour Tom,
Il fait bon aujourd'hui. Bonne semaine !
```

4.3. L'école organise un souper et propose un menu enfant (moins de 12ans) à 8€50 et un menu adulte à 14€. Écrivez un programme qui demande à l'utilisateur le nom de famille (les noms composés doivent pouvoir être encodés, max 128 caractères), le nombre d'enfants qui seront présents et le nombre d'adultes. Le dialogue se présentera de la manière suivante :

```
Bonjour, merci de participer a notre souper.

Quel est votre nom de famille ?

> Dubois

Combien d'adultes prendront part au souper ?

> 1

Combien d'enfants prendront part au souper ?

> 1
```

Le programme affichera alors le prix à payer à la famille sous la forme suivante. Veillez à afficher une réponse en nombre flottants à deux décimales de précisions.

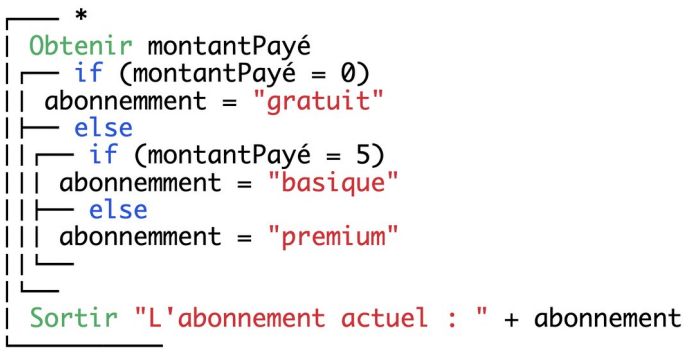
```
Reservation : Dubois

Prix a payer : 22.50 euro
```

4.4. Écrire un programme qui demande trois prénoms à l'utilisateur puis qui informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique. (On ne demande pas de restituer l'ordre alphabétique des trois prénoms)

4.5. Écris un programme qui demande à l'utilisateur de deviner un mot magique. Si la réponse est correcte, affiche "Bravo ! Vous avez trouvé le mot magique !". Sinon, affiche "Ce n'est pas ça, réessayez !".
Amélioration : ajouter plusieurs mots magiques.

4.6. Ecrire un programme qui affiche la taille d'une chaine de caractères donnée par l'utilisateur. Par exemple, pour « langage » le programme affichera `7`.

4.7. Traduis le diagramme d'action suivant en langage C :



4.8. Crée un programme qui détermine le mot le plus long parmi une liste de mots donnés par l'utilisateur. L'encodage se termine par « Q ».

5. Les tableaux

5.1. Déclare un tableau d'entiers de taille 5, initialise-le avec les valeurs `[1,2,3,4,5]` et affiche chacun de ses éléments à l'aide d'une boucle.

5.2. Déclare un tableau de réels contenant les valeurs `[2.5,3.5,4.5,5.5]`, puis calcule et affiche la somme de tous ses éléments.

5.3. Déclare un tableau de caractères contenant la chaîne `"abracadabra"`. Affiche chaque caractère de la chaîne, puis détermine et affiche le caractère le plus fréquent dans le tableau.

5.4. Déclare un tableau de booléens de taille 5, initialisé avec `[true,false,true,false,true]`. Affiche les valeurs du tableau et compte combien de fois la valeur `true` apparaît.

5.5. Ecris un programme qui demande à l'utilisateur une série de prix-quantités puis qui affiche le résultat sous la forme d'un ticket de caisse (avec le total à la fin).

5.6. Ecris un programme qui demande à l'utilisateur deux chaines de caractères puis qui concatène la première avec la deuxième. Attention, tu ne peux pas utiliser la fonction `strcat` ni concaténer directement dans l'affichage du `printf`.

5.7. Ecris un programme qui demande à l'utilisateur une série d'entiers puis qui affiche cette série à l'envers.

5.8. Ecris un programme qui vérifie si un tableau d'entiers est symétrique.

5.9. Crée un programme qui détermine si un mot est un palindrome ou non.

5.10. Ecris un programme qui demande à l'utilisateur d'insérer 10 entiers puis qui affiche la liste de ces entiers triés dans l'ordre croissant. Implémente l'algorithme de tri à bulles pour trier le tableau.

5.11. Écris un programme qui "compresse" un tableau de 0 et de 1 en déplaçant tous les 1 à gauche et tous les 0 à droite. Par exemple, le tableau `[1,0,1,0,0,1,0]` deviendrait `[1,1,1,0,0,0,0]`.

5.12. Ecris un programme qui demande une chaine de caractères à l'utilisateur puis qui remplace toutes les voyelles de cette chaine par le symbole `*`.

5.13. Ecris un programme qui demande une chaine de caractères à l'utilisateur ainsi qu'une lettre. Le programme doit se charger de remplacer toutes les occurrences de cette lettre dans la chaine par le symbole `*`.

5.14. Pour chaque problématique ci-dessous, complète la condition du while qui permettra de réaliser la recherche demandée.

Problématique	Condition
Vous avez un tableau d'entiers `nombres[]` de taille `taille`. Écrivez la condition d'une boucle while qui recherche la première occurrence de la valeur `valeurRecherchee` dans ce tableau.	
Vous avez un tableau de chaînes de caractères `noms[]` contenant `nbNoms` éléments. Écrivez la condition d'une boucle while qui recherche la chaîne `nomRecherche` dans ce tableau.	
Vous avez un tableau d'identifiants `ids[]` de taille `nbIds`. Écrivez la condition d'une boucle while qui recherche l'identifiant `idRecherche` dans ce tableau.	
Vous avez un tableau d'entiers `nombres[]` de taille `taille`. Écrivez la condition d'une boucle while qui recherche le premier nombre négatif dans ce tableau.	
Vous avez un tableau d'entiers `nombres[]` de taille `taille`. Écrivez la condition d'une boucle while qui recherche le premier nombre pair dans ce tableau.	
Vous avez un tableau de températures `temperatures[]` de taille `nbTemperatures`. Écrivez la condition d'une boucle while qui recherche la première température supérieure à `seuil`.	
Vous avez un tableau de mots `mots[]` contenant `nbMots` éléments. Écrivez la condition d'une boucle while qui recherche le premier mot commençant par la lettre `lettre`.	
Vous avez une chaîne de caractères `texte[]`. Écrivez la condition d'une boucle while qui recherche la première occurrence du caractère `caractereRecherche` dans cette chaîne.	

5.15. Écris un programme qui demande à l'utilisateur de saisir une série de nombres distincts (limités à une taille maximale de 10). Ensuite, demande-lui d'ajouter un nouveau nombre. Si ce nombre existe déjà dans le tableau, affiche un message d'erreur. Sinon, ajoute-le au tableau et affiche la liste mise à jour.

5.16. Écris un programme qui demande à l'utilisateur d'entrer 5 nombres distincts, puis un nombre à rechercher, et affiche sa position dans le tableau ou le message « Nombre introuvable » s'il n'est pas présent.

5.17. Écris un programme qui demande à l'utilisateur de saisir une série de nombres entiers. L'encodage s'arrête dès que l'utilisateur entre un nombre déjà saisi auparavant. Le programme doit ensuite afficher tous les nombres uniques saisis dans l'ordre d'entrée.

5.18. Écris un programme qui calcule et affiche la somme des éléments des deux diagonales d'une matrice carrée donnée par l'utilisateur. L'utilisateur donne dans un premier temps la taille `N` de la matrice (`N`x`N`) puis le contenu de la matrice sous la forme d'une suite d'entiers.

5.19. Ecris un programme qui demande à l'utilisateur d'entrer une chaîne de caractères et compresse cette chaîne en utilisant l'algorithme de compression RLE (Run-Length Encoding). Par exemple, la chaîne `"aaabbccccd"` deviendrait `"a3b2c4d1"`.

5.20. Ecris un programme qui demande à l'utilisateur de saisir une chaîne de chiffres (par exemple `"12345"`) puis qui convertit chaque caractère de la chaîne en un entier et qui stocke ces entiers dans un tableau.

5.21. Ecris un programme qui affiche la fréquence de chaque lettre dans une chaine de caractères entrée par l'utilisateur.

5.22. Écris un programme qui demande à l'utilisateur de saisir deux tableaux d'entiers triés (de tailles 5 chacun). Ensuite, fusionne ces deux tableaux dans un seul tableau trié en ordre croissant et affiche le résultat.

6. Les fonctions

6.1. Pour chaque programme, identifie les prototypes, les appels ainsi que les déclarations de fonctions.

```c
void f1(char p[]);

void main(void){
    f1("Texte");
}

void f1(char p[]) {
    printf("%s\n", p);
}
```

```c
int f2(int x, int y);

void main(void) {
    int r;
    r = f2(8, 3);
    printf("Résultat : %d\n", r);
}

int f2(int x, int y) {
    return x + y;
}
```

```c
void f3(void);
double f4(double a, double b);

void main(void){
    double r;
    f3();
    r = f4(2.1, 3.9);
    printf("Résultat : %.2f\n", r);
}

void f3(void) {
    printf("Message !\n");
}

double f4(double a, double b) {
    return a * b;
}
```

6.2. Pour chaque programme, identifie les paramètres effectifs et les paramètres formels.

```c
void afficherMessage(char message[]) {
    printf("Message: %s\n", message);
}

void main(void) {
    afficherMessage("Bonjour!");
}
```

```c
int somme(int a, int b) {
    return a + b;
}

void main(void) {
    int resultat = somme(10, 20);
    printf("Résultat: %d\n", resultat);
}
```

```c
void afficherDivision(int a, int b) {
    if (b != 0) {
        printf("Division: %d\n", a / b);
    } else {
        printf("Division par zéro impossible\n");
    }
}

void main(void) {
    afficherDivision(15, 3);
    afficherDivision(10, 0);
}
```

6.3. Pour chaque programme, détermine la portée (*scope*) de chaque variable.

```c
void f(int x, int y) {
    x = x + 10;
    y = y + 20;
    printf("x = %d, y = %d\n", x, y);
}

void main(void) {
    int a = 5, b = 15;
    f(a, b);
    printf("a = %d, b = %d\n", a, b);
}
```

La variable	est accessible de la ligne...	à la ligne...
x	2	4
y		
a		
b		

```c
void f() {
    int x;
    for (int i = 0; i < 3; i++) {
        x = i * 2;
        printf("x (inside loop) = %d\n", x);
    }
}

void main(void) {
    f();
}
```

La variable	est accessible de la ligne...	à la ligne...
i		
x		

```
1  void f() {
2      int a = 10;
3      printf("a (in function) = %d\n", a);
4  }
5
6  void main(void) {
7      int a = 5;
8      printf("a (in main) = %d\n", a);
9      f();
10 }
```

La variable	est accessible de la ligne...	à la ligne...
a (déclaration ligne 2)		
a (déclaration ligne 7)		

6.4. Pour chaque programme, indique si les paramètres passés à la fonction sont passés par copie ou par référence. Indique également (sans t'aider de l'ordinateur) l'affichage que produira le programme.

```
void traiter(int valeur) {
    valeur = 42;
}

void main(void) {
    int nombre = 5;
    traiter(nombre);
    printf("%d\n", nombre);
}
```

```
void traiter(int valeur) {
    valeur = 42;
    return valeur;
}

void main(void) {
    int nombre = 5;
    nombre = traiter(nombre);
    printf("%d\n", nombre);
}
```

```
void traiter(float valeur) {
    valeur = 3.14;
}

void main(void) {
    float nombre = 1.23;
    traiter(nombre);
    printf("%.2f\n", nombre);
}
```

```
void traiter(int tableau[], int taille) {
    tableau[0] = 99;
}

void main(void) {
    int valeurs[3] = {1, 2, 3};
    traiter(valeurs, 3);
    printf("%d\n", valeurs[0]);
}
```

```c
void traiter(char caractere) {
    caractere = 'Z';
}

void main(void) {
    char lettre = 'A';
    traiter(lettre);
    printf("%c\n", lettre);
}
```

```c
void traiter(int valeur) {
    valeur = 99;
}

void main(void) {
    int valeurs[3] = {1, 2, 3};
    traiter(valeurs[0]);
    printf("%d\n", valeurs[0]);
}
```

6.5. Pour chaque module, écris les prototypes des fonctions correspondantes. Enfin, indique quels sont les appels corrects et ceux qui ne le sont pas (écris « OK » ou « KO » à coté de chaque appel).

```
o————————————————————————————o
| afficherMessageDeBienvenue |
o————————————————————————————o
```

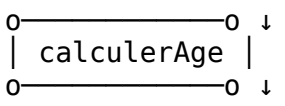
```
printf(afficherMessageDeBienvenue());
afficherMessageDeBienvenue;
afficherMessageDeBienvenue();
char msg = afficherMessageDeBienvenue();
printf("%s", afficherMessageDeBienvenue());
```

```
o————————————————————o ↓ age, statut
| afficherPrixEntree |
o————————————————————o
```

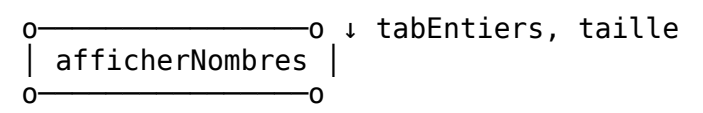
```
afficherPrixEntree();
afficherPrixEntree(17, 'E');
afficherPrixEntree('T', 17);
float prix = afficherPrixEntree(17, 'E');
```

```
o————————————————————————o
| obtenirNombreAleatoire |
o————————————————————————o ↓ nombre
```

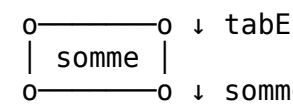
```
obtenirNombreAleatoire(3);
obtenirNombreAleatoire();
int nombre = obtenirNombreAleatoire();
printf("%d", obtenirNombreAleatoire());
int nombre = obtenirNombreAleatoire() * 2;
```



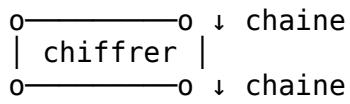

```
printf("%d", calculerAge(1998,12,23));
calculerAge(1998,12,23);
int age = calculerAge(1998,12,23);
int age = calculerAge(19 * 10, 3 + 10, 100 – 2);
```




```
int tabEntiers[] = {10, 20, 30, 40, 50};

printf("%d", afficherNombres(tabEntiers, sizeof(tabEntiers));
int x = afficherNombres(tabEntiers, sizeof(tabEntiers));
afficherNombres(tabEntiers, sizeof(tabEntiers));
afficherNombres(tabEntiers, 5);
```




```
int tabEntiers[] = {10, 20, 30, 40, 50};

printf("%d", somme(tabEntiers, sizeof(tabEntiers));
int x = somme(tabEntiers, sizeof(tabEntiers));
somme(tabEntiers, sizeof(tabEntiers));
somme(tabEntiers, 5);
```




```
char chaine[] = "Hello !";


printf("%s", chiffrer(chaine, sizeof(chaine));
chiffrer(chaine, sizeof(chaine));
chiffrer(chaine[]);
chiffrer(chaine);
```

6.6. Complète le tableau suivant

Prototype	Appel
`void afficherMessage(void);`	
	`int somme = addition(5, 7);`
`void afficherNombre(int n);`	
	`int carre = calculerCarre(4);`
`bool estPositif(int nombre);`	
	`bool estValide = verifValid('A');`
`void afficherTableau(int tab[], int taille);`	
	`float moyenne = calculerMoyenne(notes, 5);`
`int rechercherElement(int tableau[], int taille, int valeur);`	
	`char lettre = obtenirCaractere(texte, 3);`
`int compterOccurrences(char texte[], char caractere);`	
	`float resultat = puissance(2.5, 3);`
`void trierTableau(float donnees[], int debut, int fin);`	
	`int maximum = trouverMax(nombres, 10);`
`float sommeTableau(float valeurs[], int taille);`	

6.7. Écris une fonction qui retourne la valeur de la somme de cinq nombres fournis en argument. Testez cette fonction.

6.8. Écris une fonction qui renvoie le nombre de voyelles contenues dans une chaîne de caractères passée en argument. Testez cette fonction.

6.9. Écris une fonction booléenne qui retourne un booléen à true si un tableau est trié, à false si le tableau n'est pas trié. Testez la fonction. La fonction possède comme argument le tableau, la longueur du tableau.

6.10. Écrivez une fonction qui nous indique si un tableau est trié. En argument, cette méthode possède le tableau, la longueur du tableau, un booléen qui nous indique si on doit tester le tri en ordre croissant (true) ou en ordre décroissant (false).

6.11. Ecris une fonction « exposant » qui calcule l'exposant d'un nombre donné.
La fonction possède deux arguments. Le premier représente la base, le second l'exposant.
La fonction retourne en résultat le résultat de base ^{exposant}
Dans le programme principal :

- Demandez à l'utilisateur d'insérer un nombre pour la base, un nombre pour l'exposant.
- Affichez le résultat de la fonction exposant.

6.12. Ecris une fonction « palindrome », cette fonction contient un paramètre : le mot à analyser. (le mot ne contient pas d'espace.) La fonction retourne true si le mot est un palindrome, false si la fonction n'est pas un palindrome. Dans le programme principal : demandez un mot à l'utilisateur. Testez ensuite la fonction palindrome. Si elle retourne une valeur vraie, indiquez à l'écran: « le mot est un palindrome ». Si elle retourne une valeur fausse indiquez à l'écran : « le mot n'est pas un palindrome ».

6.13. La suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence par les termes 0 et 1 et ses premiers termes sont :

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	...	F_n
0	1	1	2	3	5	8	13		$F_{n-1}+F_{n-2}$

Créez une fonction « fibonacci » qui calcule la valeur du $n^{ième}$ terme de la suite. Cette fonction sera codée de manière itérative.

Le programme principal, demandera une valeur à l'utilisateur, Affichera ensuite une fois le résultat de la fonction «fibonacci».

6.14. Crée un programme qui permet de lancer les deux derniers exercices. Pour cela, créez une nouvelle fonction « interfacePrincipale » qui appelle le palindrome ou la suite de fibonacci. en fonction du choix de l'utilisateur. Si l'utilisateur entre autre chose que 1 ou 2, le programme répondra. « choix incorrect, veuillez réessayer ». À la fin de chaque exercice, introduis un extrait de code qui permette de revenir au menu principal. Voici un exemple d'interface :

```
--------------------------------
Bienvenue sur le menu principal
1. exercice palindrome ")
2. exercice fibonacci ")
--------------------------------

Choisissez le numéro de l'exercice :
```

7. Synthèse des compétences

Créez un programme qui contienne 2 tableaux.

- Le premier contiendra le nombre de tours effectués.
- Le deuxième contiendra l'écurie.

Afin de garder un cohésion entre les deux tableaux. **Les index de ceux-ci feront à chaque fois référence au numéro de la voiture**.

1 L'objectif de l'exercice sera le suivant :

- Le programme demande à l'utilisateur d'insérer un numéro de voiture.
- Si le numéro de la voiture n'est pas correct, le programme indiquera un message d'erreur.
- Si le numéro est correct, le logiciel affiche, le nombre de tours effectués ainsi que l'écurie de la voiture.
- Avant de s'interrompre, le logiciel demande à l'utilisateur s'il veut demander les informations d'une autre voiture. Réponse par O / N.

Voiture	Equipe / Ecurie	Nombre de tours
0	Undefined	-1
1	Porsche Team	346
2	Signatech Alpine	357
3	Toyota Gazoo Racing	381
4	Audi Sport Team Joest	372
5	Race Performance	297
6	G-Drive Racing	353
7	Murphy Prototypes	323
8	SO24! by Lombard Racing	328
9	Greaves Motorsport	348
10	Strakka Racing	351
11	AF Corse	179
12	Manor	283

2 L'objectif de l'exercice sera le suivant :

Le programme affichera les informations de l'ensemble des voitures sous un dialogue tel que :

```
voiture : 1 | equipe : Porsche Team | tours : 346
voiture : 2 | equipe : Signatech Alpine | tours : 357
voiture : 3 | equipe : Toyota Gazoo Racing | tours : 381
```

.....

3 L'objectif de l'exercice sera le suivant : après une recherche dans les tableaux, le logiciel affichera le podium de la course.

4 Crée une interface qui donne accès à chacun des exercices. Exemple de dialogue :

```
_--------------------------------
1. Rechercher une voiture
2. Afficher la liste des voitures
3. Afficher le podium
_--------------------------------
Choisissez un numero 1-3 : 5
Numero incorrect !
Choisissez un numero 1-3 : _
```

Adaptez les exercices afin de permettre le retour au menu principal. Exemple de dialogue
`Pour retourner au menu principal, tapez Q`

8. Les pointeurs

8.1. Observe le code suivant :

```
char u, v = 'E';
char *pu, *pv = &v;

*pv = v – 3;
u = *pv + 2;
pu = &u + 1;
```

Sachant que la variable `u` se trouve à l'adresse `A1B` et v à l'adresse `A1C`, que valent :

- `*pv ?`
- `u ?`
- `pu ?`
- `*pu ?`

8.2. Observe le code suivant :

```
float a = 12.4;
float b = -3.6;
float c;
float *pa = &a
float *pb = &b;
```

Sachant que a, b et c se trouvent respectivement aux adresses 2471, 2138 et 1998, quel sera l'effet des instructions suivantes :

- `*pa = 5*a`
- `c = 2*(*pa - *pb)`
- `--*pb`
- `(*pa)++`
- `pa++`

8.3. Écris une fonction nommée « initialisation » qui se charge d'initialiser trois variables entières (n1, n2, n3) à 0.

8.4. Observe le code suivant :

```
int tabNbr[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
int *pTabNbr =  tabNbr;
int i = 2;
```

Pour chaque expression, indique son type et sa valeur ainsi que la valeur de `pTabNbr` après l'exécution.

Expression	Type de l'expression	Valeur de l'expression	Valeur de `pTabNbr` après exécution
`pTabNbr`			
`*pTabNbr`			
`++pTabNbr`			
`(*pTabNbr)--`			
`*(++pTabNbr)`			
`pTabNbr + i`			
`(*pTabNbr + i)`			
`*(i + pTabNbr)`			

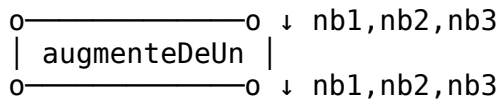
8.5. Écris une fonction qui calcule la moyenne des éléments d'un tableau d'entiers sachant que son prototype est :

```
float moyenne (int *pTab, int taille);
```

Ensuite, écris la fonction `main` dans laquelle tu déclares un tableau de 5 notes (entières) ainsi qu'une variable `moyenneNotes`. Fait en sorte que cette variable contienne le résultat de la fonction `moyenne`.

8.6. Écris une fonction qui reçoit une chaîne de caractères et qui renvoie `true` si cette chaîne contient au moins un chiffre et `false` sinon. Attention, tu ne peux ni utiliser la fonction `strlen` ni les constantes symboliques, ni les crochets (`[]`).

8.7. Quels sont l'appel et le prototype de la fonction suivante ?



8.8. Observe le code suivant :

```
char chaine[] = "l'INDBG c'est cool !";
char *p = chaine + 8;
```

Quel sera l'affichage produit par chacune des instructions **successives** suivantes ?

- `printf("%c\n",*p);`
- `printf("%s\n", ++p);`
- `printf("%c\n",++(*p));`
- `printf("%s\n", p);`
- `printf("%c\n",*--p);`
- `printf("%c\n",*p+2);`
- `printf("%c\n",*p);`

8.9. Ecris une fonction qui ajoute un tableau de réels à la suite des réels déjà présents dans un autre tableau. Le premier tableau est suffisamment long pour y ajouter le second tableau. Le prototype est :

```
void concatenation(float *pTab1, float *pTab2, int tailleTab1, int
tailleTab2);
```

8.10. Ecris une fonction qui permet de connaître le nombre d'occurrences de chacune des lettres de l'alphabet dans un texte.

9. Les structures

9.1. On te demande de concevoir un programme permettant l'encodage de plusieurs élèves. Pour chaque élève, on souhaite retenir son nom, son prénom, son age, son sexe ainsi que sa moyenne globale obtenue durant chaque année scolaire.

a. Ecris la structure correspondante.

b. Déclare un type (synonyme) pour cette structure.

c. Déclare et initialise une variable nommée `jean` de ce type contenant les informations de Jean Dubois (de ton choix).

d. Réécris la structure (ajoute des nouvelles structures et synonymes si c'est nécessaire) : pour chaque année scolaire, on souhaite retenir la moyenne obtenue ainsi que l'identifiant de sa classe (ex. « 4G1 », « 5G2 », etc.)

e. Ecris une fonction qui se charge d'afficher les informations d'un élève (reçu en argument) de la manière suivante (les parties soulignées doivent varier en fonction des infos de l'élève) :

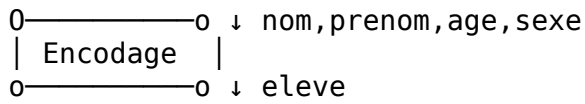
```
Fiche de l'élève : Jean Dubois (H)
- 1C : 70/100
- 2C : 65/100
- 3G1 : 78/100
- 4G2 : 73/100
```

f. Déclare un tableau contenant 5 élèves de ton choix et pouvant contenir 10 élèves au maximum.

g. Ecris une fonction qui se charge d'ajouter un nouvel élève (reçu en argument) dans le tableau d'élèves (reçu en argument).

h. Ecris l'instruction permettant d'enregistrer une nouvelle moyenne à Jean : en 5G1, il a obtenu 68/100.

9.2. On souhaite créer un programme permettant l'encodage d'un élève. On te demande d'écrire la fonction correspondante au module suivant :



La fonction encodage prend en entrée une série d'information sur l'élève et retourne une structure

9.3. On souhaite gérer des produits dans un entrepôt. Un produit est défini par 3 critères:

- l'entier code produit ("567" par exemple qui est un nombre entre 255 et 450),
- l'intitulé ("pots de peinture" et qui comporte au maximum 99 caractères utiles)
- un entier qui indique la quantité en stock (803 par exemple).

Le tableau des produits comportera au maximum 100 produits.

Il faut gérer le tableau grâce au menu suivant :

1. Ajouter un produit (on tape le code produit et l'intitulé, la quantité et le stock).

2. Afficher la liste de produits.
3. Modifier un élément existant.

On veillera à bien décomposer ce problème en différents modules et à mener une réflexion sur les fonctions nécessaires dans chaque module.

Par défaut, on estime que les insertions faites par l'utilisateur sont correctes.

Dépassement : Il faut gérer une liste de produits en veillant à ce qu'il n'y ait pas deux produits avec le même code produit. Bien sûr la quantité en stock ne peut pas être négative.

L'interface principale renvoie vers trois procédures distinctes :

- menuAjouter
- menuResume
- menuModifier

Celles-ci doivent parfois également utiliser des sous-fonctions. Notamment pour la recherche d'indice dans menuModifier.

10. Les fichiers

10.1. Tu est responsable de la base de données des produits d'une grande surface. Un article est caractérisé par un libellé (ex. « Lait entier ») et un prix (ex. 1,50€). Les articles doivent être encodés dans un fichier binaire (enregistrements de longueur fixe).

a) Déclare la structure correspondante

b) Déclare un pointeur permettant d'avoir **accès en écriture** aux informations enregistrées dans le fichier

c) Ajoute plusieurs articles :

i. « Croissant » au prix de 0,50€

ii. « Lait entier » au prix de 1,20€

iii. « Compote » au prix de 2,20€

iv. « Pain complet » au prix de 2,50€

d) Ferme le fichier puis rouvre-le afin d'y avoir **accès uniquement en lecture**

e) Affiche la liste des articles

f) Ferme le fichier puis rouvre-le afin d'y avoir **accès en lecture et écriture** (sans écraser le fichier)

g) Modifie le prix de la compote pour qu'il soit de 2,50€

h) Ferme le fichier puis rouvre-le afin d'y avoir **accès uniquement en lecture**

i) Affiche la liste des articles et vérifie que le prix de la compote a bien été modifié

10.2. Écris un programme qui crée un fichier `avis_restaurants.dat` qui reprend des données sur des restaurants. Par restaurant, nous souhaitons connaître le nom du restaurant (max. 100 caractères), le nombre d'avis donnés (un entier) ainsi que la moyenne des notes attribuées de 0 à 5 (un réel). Chaque enregistrement est de longueur fixe. Ton programme doit fournir 2 fonctionnalités (sous forme d'un menu) :

a) la création du fichier avec l'encodage de plusieurs restaurants (le nombre d'avis et la moyenne des notes doivent être initialisées à 0 automatiquement)

b) l'affichage des restaurants sous la forme
<nom> (<moyenne>/5 - <nbAvis> avis)

10.3. Ajoute une nouvelle fonctionnalité au programme réalisé dans l'exercice précédent : l'utilisateur doit pouvoir ajouter une note à un restaurant existant. La moyenne doit alors être automatiquement recalculée et mise à jour. Le nom du restaurant peut ne pas être dans le fichier : dans ce cas, il faut afficher un message d'erreur. Par facilité, les entrées de l'utilisateur (nom du restaurant et note) doivent être stockées dans une structure.

10.4. Ajoute trois nouvelles fonctionnalités :

a) Ajouter un restaurant à la fin du fichier (sans écraser le fichier existant)

b) Supprimer un restaurant (par facilité, le nom est simplement remplacé par « *** » ; l'enregistrement est encore physiquement présent dans le fichier). Attention, fais en sorte que les restaurants supprimés ne s'affichent pas dans la liste des restaurants

c) Modifier le nom d'un restaurant