

Université de Nice-Sophia Antipolis

Licence professionnelle

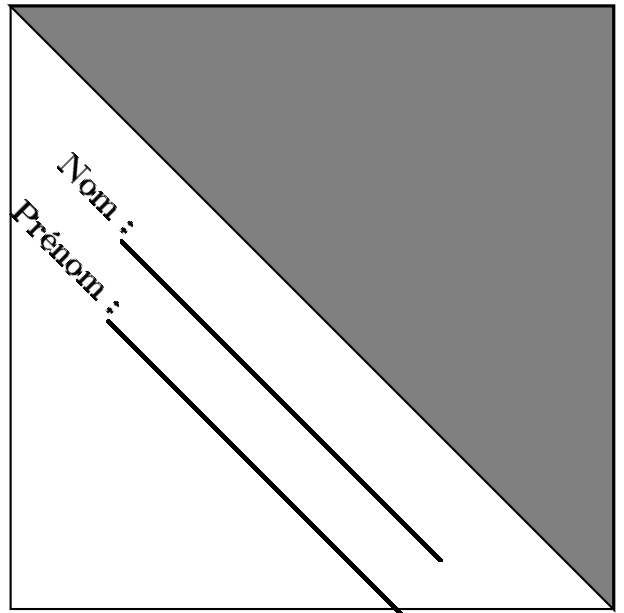
2002–2003

Examen de Langage C

Durée : 1h30

Aucun document autorisé

Note :



1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Exercice 1

1. Définissez un nouveau nom de type `Couple` représentant un type structuré comportant deux champs, `epoux` et `epouse`, qui sont des chaînes de caractères pouvant contenir des mots de longueur 19 au plus.

2. Complétez le fragment de code de la page suivante afin de définir un tableau de `Couple` de taille 2, et de remplir chacun des éléments de ce tableau. Vous donnerez leur valeur aux champs d'un élément du tableau en utilisant les chaînes de caractères définies localement.
Si vous ne reconnaissez pas les couples célèbres, vous pouvez les accorder aléatoirement !

```
int main(void) {
    char bonaparte []="Bonaparte";
    char ulyse []="Ulysse";
    char josephine []="Joséphine";
    char penelope []="Pénélope";
```

Exercice 2

3. Écrivez une fonction `trinôme` calculant les racines de $f(x) = ax^2 + bx + c$. La fonction prend en paramètre un tableau de réels contenant les coefficients a , b et c , et deux autres paramètres passés par référence, `r1` et `r2`, qui lui permettront de communiquer la valeur des racines à la fonction appelante. Enfin, la fonction renvoie la valeur du discriminant.

Notes : Le discriminant est $\Delta = b^2 - 4ac$.

Si $\Delta < 0$, il n'y a pas de solution, les valeurs de r_1 et r_2 sont indéterminées.

Si $\Delta = 0$, il y a une solution $r_1 = \frac{-b}{2a}$ et la valeur de r_2 est indéterminée.

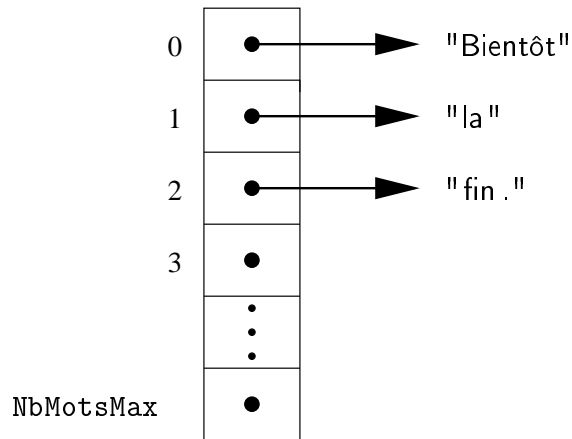
Si $\Delta > 0$, il y a deux solutions $r_1 = \frac{-b - \sqrt{\Delta}}{2a}$ et $r_2 = \frac{-b + \sqrt{\Delta}}{2a}$.

Exercice 3

Le but de cet exercice est de découper une chaîne de caractères en mots. On appellera *mot* toute suite de caractères délimitée par des blancs. Un caractère *blanc* est ici pris au sens large et pourra être identifié à l'aide de la fonction `isspace` de la bibliothèque standard.

Les mots de la chaîne initiale seront stockés dans un tableau de chaînes de caractères à la manière du paramètre `argv` de la fonction `main`. La taille d'une chaîne de caractères dans le tableau dépend donc de la longueur du mot qu'elle servira à stocker. En revanche, la taille de ce tableau est fixée à `NbMotsMax`. Exemple :

"_Bientôt_la fin ." →



L'idée qui va être développée ici consiste à déplacer un pointeur sur les caractères de la chaîne initiale et de le faire progresser de mot en mot en réalisant une copie de chacun des mots au passage. Grossièrement décrit, l'algorithme se résume à sauter d'éventuels blancs, à calculer la longueur du mot courant, puis à créer la chaîne de

source dans *destination*. La fonction renvoie un pointeur sur le premier caractère suivant le mot copié dans *source*.

```
char *copierMot(char *source, char *destination) {
```

9. Enfin, écrivez une fonction `main` dans laquelle vous définirez un tableau de chaînes de caractères (selon le sens donné dans le préambule de l'exercice) et une chaîne de caractères initialisée (à la définition) par une phrase de votre choix. Vous utiliserez alors les fonctions définies précédemment (et seulement celles-là!) pour remplir le tableau de chaînes de caractères avec tous les mots de votre phrase.

Exercice 4

```
int main(void) {
    int *p1, *p2, temp, t[]={0,2,4,6,8};

    p1 = t;
    p2 = p1+4;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;

    while(*p1) {
        *p1++=p2[temp];
        temp--;
    }
    return 0;
}
```

10. Complétez les lignes du tableau suivant en notant la valeur des variables du programme après chaque instruction. Pour les pointeurs, vous représenterez leur valeur en utilisant le symbole `&`, suivi du nom de la variable pointée ou de la case dans le tableau `t` (e.g. `&t[3]`). Une valeur indéterminée est représentée par `?`.

main	t[0]	t[1]	t[2]	t[3]	t[4]	p1	p2	temp
Initialisations	0	2	4	6	8	?	?	?
<code>p1 = t;</code>								
<code>p2 = p1+4;</code>								
<code>temp = *p1;</code>								
<code>*p1 = *p2;</code>								
<code>*p2 = temp;</code>								

11. De la même façon et à la suite, indiquez la valeur des données du programme **après** chaque tour de boucle. Le nombre de lignes du tableau n'est pas significatif.

<pre>while(*p1) { *p1++=p2[temp]; temp--; }</pre>	t[0]	t[1]	t[2]	t[3]	t[4]	p1	p2	temp

Annexe

Tableau des opérateurs rangés par priorité décroissante :

Types	Symboles	Associativité
postfixé	() , [] , . , -> , ++ , --	G → D
unaire	& , * , + , - , ~ , ! , ++ , -- , sizeof	D → G
casting	(<type>)	D → G
multiplicatif	* , / , %	G → D
additif	+ , -	G → D
décalage	<< , >>	G → D
relationnel	< , > , <= , >=	G → D
(in)égalité	== , !=	G → D
et bit à bit	&	G → D
xor	^	G → D
ou bit à bit		G → D
et logique	&&	G → D
ou logique		G → D
condition	? :	D → G
affectation	= , *= , /= , %= , += , -= , <<= , >>= , &= , ^= , =	D → G
virgule	,	G → D

Extraits du *man* d'UNIX :

`int isspace (int c);`

`isspace` vérifie si l'on a un caractère blanc, d'espacement. Il s'agit de: espace, saut de page (form-feed, `'\f'`), saut de ligne (newline, `'\n'`), retour chariot (carriage return, `'\r'`), tabulation horizontale (`'\t'`), et tabulation verticale (`'\v'`).

La valeur renvoyée est non nulle si le caractère `c` entre dans la catégorie testée, et zéro sinon.

`char *strcpy (char *dest, const char *src);`

La fonction `strcpy()` copie la chaîne pointée par `src` (y compris le caractère `'\0'` final) dans la chaîne pointée par `dest`. Les deux chaînes ne doivent pas se chevaucher. La chaîne `dest` doit être assez grande pour accueillir la copie.

La fonction `strcpy()` renvoie un pointeur sur la chaîne destination `dest`.

`double sqrt (double x);`

La fonction `sqrt()` renvoie la racine carrée non-négative de `x`. Elle échoue et positionne `errno` à `EDOM`, si `x` est négatif.

`void *malloc (size_t size);`

`malloc()` alloue `size` octets, et renvoie un pointeur sur la mémoire allouée. Le contenu de la zone de mémoire n'est pas initialisé.

La valeur renvoyée est un pointeur sur la mémoire allouée, qui est correctement alignée pour n'importe quel type de variable, ou `NULL` si la demande échoue.