

<b>Travaux Dirigés - Langage C</b> <b>Série 1</b>
--

### Concaténation de chaînes

Écrivez un programme qui :

1. déclare deux chaînes de caractères (tableau de caractères) `nom` et `extension` de taille 255 caractères maximum,
2. demande à l'utilisateur « entrez le nom du fichier »
3. stocke le résultat dans `nom`
4. demande à l'utilisateur « entrez l'extension du fichier (3 caractères) »
5. stocke le résultat dans `extension`
6. concatène la chaîne `nom` et la chaîne constante "."
7. concatène la chaîne `nom` et la chaîne `extension`
8. affiche à l'écran « le fichier sera enregistré sous le nom : » suivi de la chaîne de caractères `nom`

### Calcul du module d'un nombre complexe

Écrivez un programme qui :

1. affiche à l'écran «  $z = a + ib$  »
2. demande à l'utilisateur de saisir `a` puis `b` et les stocke dans les chaînes `chaine_a` et `chaine_b`
3. calcule le module de `z` et le stocke dans la variable `module`

## Annexes

### **string.h**

**strcat(s,ct)** concatène la chaîne `ct` à la suite de la chaîne `s` et stocke le résultat dans `s`

### **math.h**

**sin(x)** sinus de `x`.

**cos(x)** cosinus de `x`.

**tan(x)** tangente de `x`.

**asin(x)** arc sinus de `x`, dans l'intervalle  $[-\pi/2, \pi/2]$ ,  $x \in [-1, 1]$

**acos(x)** arc cosinus de `x`, dans l'intervalle  $[0, \pi]$ ,  $x \in [-1, 1]$

**atan(x)** arc tangente de `x`, dans l'intervalle  $[-\pi/2, \pi/2]$

**atan2(y,x)** arc tangente de  $y/x$ , dans l'intervalle  $[-\pi, \pi]$

**sinh(x)** sinus hyperbolique de x.

**cosh(x)** cosinus hyperbolique de x.

**tanh(x)** tangente hyperbolique de x.

**exp(x)** fonction exponentielle de x.

**log(x)** logarithme népérien :  $\ln(x), x > 0$

**log10(x)** logarithme à base 10 :  $\log_{10}(x), x > 0$

**pow(x,y)**  $x^y$

**sqrt(x)**  $\sqrt{x}, x \geq 0$

**ceil(x)** le plus petit entier supérieur ou égal à x, exprimé en double

**floor(x)** le plus grand entier inférieur ou égal à x, exprimé en double

**fabs(x)** valeur absolue de x

## **stdlib.h**

**atof(chaine)** convertit chaine en un double

**atoi(chaine)** convertit chaine en un entier

**atol(chaine)** convertit chaine en un long