

Ce document est sous licence CC-by-sa, merci de la respecter.
Voir : <https://creativecommons.org/licenses/by-sa/3.0/fr/>

But de ce document : permettre à ceux qui le souhaitent, d'utiliser l'API saisie (nombres / chaînes de caractères) fournie dans leurs programmes (réalisés en projet ou ailleurs).

Introduction : la saisie de chaînes de caractères et la saisie de nombres (entier court, entier long, réel ou en notation scientifique) n'est pas une chose facile en langage C. C'est la raison pour laquelle vous disposez d'une API dédiée et simplifiée, qui se compose des fichiers présentés ci-après.

Fichiers à utiliser :

Fichiers d'en-tête, **à placer dans le répertoire "inc" de l'arborescence de votre projet** :
saisie_chaine.h // définition de la fonction saisie_chaine_de_caracteres
saisie_nombre.h // définitions des fonctions utilisables pour saisir un nombre réel ou entier

Fichiers à compiler, **à placer dans le répertoire "sources" de l'arborescence de votre projet** :
saisie_chaine.c // contient l'implémentation de la fonction saisie_chaine_de_caracteres ()
saisie_nombre_reel.c // contient l'implémentation des fonctions de saisie de nombres réels
saisie_nombre_entier.c // contient l'implémentation des fonctions de saisie de nombres réels

Description de l'interface

SAISIE DE CHAÎNES DE CARACTÈRES

Variables et fonctions disponibles :

g_buf[] : tableau de caractères de taille MAX_BUFFER_SIZE (qui vaut 512 par défaut).

Ce tableau de caractères contient la chaîne saisie par l'utilisateur. Comme sa portée est globale, on peut récupérer le contenu dans un tableau, localement (par rapport à l'appel de la fonction).

saisie_chaine_de_caractères (BOOL, BOOL) : fonction qui réalise la saisie via un appel à lit_entree().

lit_entree() : fonction interne faisant un appel à fgets() et à clean_string() et stocke la chaîne entrée et nettoyée des caractères \n dans g_buf[]

initialize_string(char *) : fonction qui permet de remplir de zéros le tableau dont on donne l'adresse en paramètre.

SAISIE DE NOMBRES

Variables et fonctions disponibles :

saisie_nombre_entier_court(BOOL, BOOL, min, max) : renvoie le nombre entier court saisi par l'utilisateur, ce nombre étant compris entre min et max (définis lors de l'appel)

saisie_nombre_entier_long(BOOL, BOOL, min, max) : renvoie le nombre entier long saisi par l'utilisateur, ce nombre étant compris entre min et max (définis lors de l'appel).

saisie_nombre_reel(BOOL, BOOL, min, max) : renvoie le nombre réel saisi par l'utilisateur, ce nombre étant compris entre min et max (définis lors de l'appel).

saisie_nombre_scientifique((BOOL, BOOL, min, max) : renvoie le nombre réel saisi par l'utilisateur, ce nombre étant compris entre min et max (définis lors de l'appel).

Pour utiliser ces fonctions directement dans vos programmes :

1. ajouter les fichiers d'en tête dont vous avez besoin dans le fichier nommé " fichier .c", c'est à dire où vous utiliserez ces fonctions.

2. ne pas oublier de placer **saisie_chaine_caracteres.c** et **saisie_nombreentier.c**, et/ou **saisie_nombre_reel.c** dans le répertoire " sources ", ainsi que **saisie_chaine.h** + **saisie_nombre.h** dans le répertoire "inc" de l'arborescence de votre projet (le Makefile compilera automatiquement ces programmes).

EXEMPLES D'UTILISATION

EXEMPLE 1 : saisie d'une chaîne de caractères

But : lire une chaîne de caractères entrée par l'utilisateur, afin de pouvoir l'utiliser dans le fichier programme.c

1. ajouter les lignes suivantes dans le fichier nommé programme.c :

```
#include <string.h>
#include "saisie_chaine.h"
#include "saisie_nombre.h"
```

2. écrire la ligne ci-dessous pour définir un tableau de caractères dans lequel sera recopié la chaîne entrée :

```
char MotSaisi[ MAX_BUFFER_SIZE ];
```

Remarque : MAX_BUFFER_SIZE vaut 512 par défaut (on peut modifier sa valeur dans saisie_chaine.h)

3. dans le .c, à l'endroit où l'on souhaite utiliser la saisie, écrire :

```
saisie_chaine_de_caracteres( FALSE, FALSE ); // voir saisie_chaine.h pour les paramètres
strcpy( MotSaisi, g_buf); // recopie ce que contient g_buf dans le tableau MotSaisi
```

```
/* Exemples pour illustrer l'utilisation de la chaîne saisie */
```

```
fprintf( stdout, "Vous avez saisi : %s \n" MotSaisi);
```

```
fprintf( stdout, "La deuxième lettre du mot saisi est : %c \n" MotSaisi[1] );
```

EXEMPLE 2 : initialisation d'une chaîne de caractères (= tableau de caractères) :

```
/* soit un tableau de caractères nommé une_chaine_de_caracteres, de taille 50 */
```

```
#define TAILLE_TABLEAU 50
```

```
char une_chaine_de_caracteres[TAILLE_TABLEAU];
```

```
/* pour remplir ce tableau de zéros, il suffit d'écrire les lignes suivantes : */
```

```
#include "saisie_chaine.h"
```

```
/* le prototype est : void initialize_string( char *); // paramètre = une adresse
```

```
Rappel : une_chaine_de_caracteres est l'adresse du premier élément du tableau */
```

```
initialize_string(une_chaine_de_caracteres);
```

EXEMPLE 3 : saisie d'un nombre entier compris entre -65535 et +65535 ("short int")

1. ajouter les lignes suivantes dans le .c où vous souhaitez saisir ce type de nombre :

```
#include "saisie_chaine.h"  
#include "saisie_nombre.h"
```

2. écrire les lignes suivantes :

```
short int min = (une valeur entière);  
short int max = (une autre valeur supérieure à la première);  
short int mon_entier_court = saisie_nombre_entier_court( FALSE, FALSE, min, max);
```

=> c'est tout !

L'utilisation des autres fonctions (saisie entier long, nombre réel, nombre en notation scientifique est complètement similaire).

Merci d'avance pour tout retour et/ou suggestion constructive !

Eric Bachard 26 mai 2016