

**Fonction de lecture et validation
d'une chaîne de caractères sur *stdin***

Fonction litEtValideChaine()

```
/*
 * PRE: inv : chaine de caracteres
 *      s : tableau de char de taille sz
 * POST: La fonction a affiche le message d'invitation inv sur stdout,
 *        ensuite elle a lu sur stdin une chaine de caracteres d'au
 *        plus sz-1 caracteres, a verifie qu'elle n'est ni vide,
 *        ni trop longue et repete l'operation tant qu'une chaine
 *        valide n'est pas introduite, puis il a remplace le '\n'
 *        par '\0'. La chaine valide a ete placee dans s.
 * RES: renvoie le nombre de caracteres de la derniere chaine lue ;
 *      -1 si fin de fichier [Ctrl-D] atteinte
 */
int litEtValideChaine (char* inv, char* s, int sz)
```

Fonction litEtValideChaine()

```
int litEtValideChaine (char* inv, char* s, int sz)
{
    // affichage de l'invite de lecture
    printf("%s", inv);

    while (fgets(s, sz, stdin) != NULL) {

        // traitement d'une ligne vide

        // traitement d'une ligne trop longue

        // chaine valide: suppression de '\n' et
        // renvoi du nombre de caracteres de la chaine

    }

    // traitement de EOF
}
```

Traitement des cas
de lignes non valides

Fonction litEtValideChaine()

```
int litEtValideChaine (char* inv, char* s, int sz)
{
    // affichage de l'invite de lecture
    printf("%s", inv);

    while (fgets(s, sz, stdin) != NULL) {

        // traitement d'une ligne vide
        if (strlen(s) == 1)
            continue;

        // traitement d'une ligne trop longue
        if (s[strlen(s)-1] != '\n') {
            // vidage du buffer de lecture
            ...
            continue;
        }

        // chaine valide: suppression de '\n' et renvoi
        // du nombre de caracteres de la chaine
        s[strlen(s)-1] = '\0';
        return strlen(s);
    }

    // traitement de EOF
    return -1;
}
```

Vidage du buffer de lecture

- vidage du buffer de lecture bloc par bloc :

```
while (fgets(s,sz,stdin) && s[strlen(s)-1] != '\n');
```

- vidage du buffer de lecture caractère par caractère :

```
while (getchar() != '\n');
```

Programme principal

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define INVIT_SIZE 60
#define BUF_SIZE 16

int litEtValideChaine (char* inv, char* s, int sz);

int main () {
    char invite[INVIT_SIZE];
    sprintf(invite,
        "Entrez une phrase d'au plus %d caracteres:\n", BUF_SIZE-2);

    char chaine[BUF_SIZE];
    int count = litEtValideChaine(invite, chaine, BUF_SIZE);
    if (count != -1)
        printf("Chaine de %d caracteres: '%s'\n", count, chaine);
}
```