

Solution exercice 1.12

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

void clear_stdin(void);

int main(void) {
    int n; char c;

    printf("%s", "Donnez un nombre entier et un caractere : ");
    if (scanf("%d %c", &n, &c) == 2)
        printf("n = %d, c = %c\n", n, c);
    clear_stdin();

    return EXIT_SUCCESS;
}

void clear_stdin(void) {
    int c;
    do {
        c = getchar();
    } while (c != '\n' && c != EOF);
}
```

N.B. `clear_stdin` ne fonctionne que si le buffer n'est pas initialement vide.
Une autre variante, fonctionnant dans tous les cas, consisterait à écrire :

```
void clear_stdin(void) {
    fseek(stdin, 0, SEEK_END);
}
```

Solution exercice 1.13

- 1) Le programme ne fonctionne pas correctement si le nom saisi contient des espaces ... car scanf cesse de lire dès lors qu'un espace est rencontré.
- 2) Dans scanf on a écrit "%20s", avec 20 "en dur"... quand bien même on a défini dans le code une constante symbolique pour cela. Serait beaucoup mieux de faire en sorte de construire le descripteur de format du scanf de telle sorte que si la constante symbolique venait à être modifiée le descripteur de format soit automatiquement corrigé en conséquence.
- 3) Le buffer de lecture n'est pas vidé.

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

#define TAILLE_NOM 20

void clear_stdin(void);

int main(void) {

    char nom[TAILLE_NOM + 1];
    char format[10];

    sprintf(format, "%%%d^\n", TAILLE_NOM);
    printf("Entrez votre nom (%d caract. max) : ", TAILLE_NOM);
    scanf(format, nom);
    clear_stdin();

    printf("Votre nom est \"%s\"\n", nom);

    return EXIT_SUCCESS;
}

void clear_stdin(void) {
    int c;
    do {
        c = getchar();
    } while (c != '\n' && c != EOF);
}
```

N.B. La solution ci-dessus n'est pas parfaite... car elle ne supprime pas les éventuels "whitespace characters" superflus présents en fin de chaîne. Pour pallier à cela, il faudrait "trimmer" la chaîne *nom*.