

bufferisation.c

```

1 #include <stdio.h>
2
3 int main(void){
4     int n = 0;
5     printf(" toto\n");
6     n = 1/n;
7     return 0;
8 }

```

bufferisation2.c

```

1 #include <stdio.h>
2
3 int main(void){
4     int n = 0;
5     printf(" toto");
6     fprintf(stderr, " titi");
7     n = 1/n;
8     return 0;
9 }

```

lecture_entree_standard.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5     char c, d;
6     int n;
7
8     printf("c = ");
9     scanf("%c", &c);
10    printf("d = ");
11    scanf("%c", &d);
12    printf("n = ");
13    scanf("%d", &n);
14
15    printf("c lu: %c\n", c);
16    printf("d lu: %c\n", d);
17    printf("n lu: %d\n", n);
18
19    return 0;
20 }

```

lecture_chaine.c

```

1 #include <stdio.h>
2
3 int main(void){
4     char str[10];
5     char c;
6
7     //scanf(" %s", str); // dangereux si l'utilisateur entre plus de 9 caractères
8     scanf("%9s", str); // plus prudent: seuls 9 caractères sont lus
9
10    scanf("%c", &c);
11    printf("c = %c\n", c); // c contiendra le 10e caractère écrit
12
13    return 0;
14 }

```

trop_d_ouvertures.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5     for(int i=0; i<1024; i=i+1){
6         if(fopen("toto", "r") == NULL){
7             printf("i=%d\n", i);
8         }
9     }
10
11     return 0;
12 }

```

pas_d_ecriture.c

```

1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void){
5     FILE *f = fopen("toto", "w");
6
7     fprintf(f, "ceci ne sera pas ecrit dans le fichier");
8
9     _exit(0); // permet de sortir sans vider les flux, on pourrait mettre 1 / 0 à la place
10 }

```

cat.ml

```

1 let affichage fichier = let instream = open_in fichier in
2     try
3         while true do
4             let lu = input_line instream in
5             print_endline lu
6         done
7     with End_of_file -> close_in instream

```