

# ERREURS CLASSIQUES EN C

2012-10-01

© Emmanuel Chieze  
Département d'informatique, UQAM  
INF3135

## Erreurs classiques en C

2

- Utilisation d'une valeur à la place d'une adresse ou réciproquement

```
int n;  
scanf("%d", n);  
  
□ au lieu de  
scanf("%d", &n);
```

## Erreurs classiques en C

3

- Utilisation d'une valeur à la place d'une adresse ou réciproquement

```
dictionnaire_lit_entree(dictionnaire
    d, char *cle, Definition *def);
dictionnaire dico;
dictionnaire_lit_entree(&dico, a[0],
    (Definition *) &definition);
```

- au lieu de

```
dictionnaire_lit_entree(dico, a[0],
    (Definition *) &definition);
```

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

4

- Longueur inappropriée pour une chaîne de caractères

- ▣ `char s[50];` ne permet de stocker que 49 caractères effectifs (plus `'\0'`)
- ▣ `length(s)` retourne le nombre de caractères effectifs (excluant `\0`)

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

5

### ❑ Erreur de format (printf, scanf)

```
int n;
```

```
double d;
```

```
...
```

```
printf("%d %f", d, n);
```

### ❑ au lieu de

```
printf("%f %d", d, n);
```

© Emmanuel Chieze, Département d'Informatique, UQAM. INF3135 2012-10-01

## Erreurs classiques en C

6

### ❑ Variable ou zone mémoire non initialisée

- ❑ initialiser chaque variable déclarée (au moment de sa définition, ou au plus tard avant la première utilisation qui en est faite)
- ❑ initialiser chaque zone mémoire retournée par malloc

© Emmanuel Chieze, Département d'Informatique, UQAM. INF3135 2012-10-01

## Erreurs classiques en C

7

- Ne pas utiliser une zone mémoire qui a été libérée !

```
char *msg(int n, char *s)
{
    char buf[100];
    sprintf(buf, "error %d: %s\n", n, s);
    return buf;
}
```

- Problème :

- l'espace pour buf est alloué lors de sa définition
- mais c'est une variable locale à msg() : la variable et la zone mémoire associées disparaissent lors du retour de la fonction

- Solution :

- laisser le soin à la fonction appelante d'allouer l'espace
  - void msg(int n, char \*s, char \*msg)
- ou allouer l'espace par un malloc dans la fonction

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

8

- Ne pas utiliser une zone mémoire qui a été libérée !

```
for (p = listp; p != NULL ; p = p->next)
    free(p);
```

- Problème :

- p = p->next est évaluée après free(p)

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

9

- Problèmes de dépassement de valeurs
  - ▣ `fact(15)` ne plante pas mais génère une valeur aberrante
- Problèmes de dépassement de limites en ressources
  - ▣ Zone mémoire non libérée alors qu'elle n'est plus utilisée
  - ▣ Fichier non fermé alors qu'il n'est plus utilisé

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

10

- Même zone mémoire utilisée par un module et par le programme appelant
  - ▣ le programme appelant peut modifier le contenu de la zone sans que le module ne le sache
  - ▣ problème si le module est supposé contrôler cette zone
  - ▣ le module doit plutôt copier la zone mémoire concernée pour en être maître
  - ▣ cf. `dictionnaire_insere_entree()`

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01

## Erreurs classiques en C

11

### □ Problèmes liés aux opérateurs

#### ▣ problème de précedence

```
if (x & 1 == 0)
```

#### ■ équivaut à

```
if (x & (1 == 0))  -- TOUJOURS FAUX
```

#### ▣ utilisation de = au lieu de ==, ou inversement

```
while ((c==getchar()) != EOF)
    if (c = '\n')
        break;
```

© Emmanuel Chieze, Département d'Informatique, UQAM. INF3135 2012-10-01

## Erreurs classiques en C

12

### □ Problèmes liés aux opérateurs

#### ▣ utilisation de ++/-- dans des expressions complexes

#### ■ cela peut être utile de mettre les ++/-- dans des lignes à part

#### ■ et de scinder les expressions trop complexes en plusieurs expressions plus simples

© Emmanuel Chieze, Département d'Informatique, UQAM. INF3135 2012-10-01

# Références

13

- La plupart des exemples proviennent de :
  - ▣ Kernighan B.W. et Pike R. 1999. *The Practice of Programming*, Addison-Wesley

© Emmanuel Chieze, Département d'Informatique, UQAM.  
INF3135 2012-10-01