TD3 – Qualificatifs *static* et *extern*

Langage C (LC4)

semaine du 11 février

1 La portee des variables

Que se passe-t-il lorsque l'on tente de compiler le code suivant ? Justifiez votre réponse.

```
float A;
int B, C[10];

void P1 (int I, int J) {
   B= I + J;
}

void P2 () {
   int B, D;
   P1 (B, D);
}

int main ( ) {
   P1(C[5], C[6]);
   P2();
}
```

2 Utilisation de `static` sur les fonctions

Question 1. Définissez une fonction `void echange(int *c1, int *c2)` qui prend en argument deux pointeurs vers des entiers et échange les valeurs vers lesquelles ils pointent Définissez une fonction `void tri3int(int *c1, int *c2, int *c3)` qui prend en argument trois pointeurs vers des entiers et qui les trie dans l'ordre croissant, en ce sens que le pointeur `c1` pointera vers le plus petit entier et `c3` vers le plus grand. (Il s'agit d'utiliser la fonction `echange`.)
Solution:

```
void echange(int *c1, int *c2)
{
  int temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
```

```
}

void tri3int(int *c1, int *c2, int *c3)
{
  if (*c1> *c2)
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
}
```

On suppose que ces fonctions appartiennent à un fichier `tri3int.c`.

Question 2. De manière analogue définissez une fonction **void** `echange(`**char** `*c1,` **char** `*c2)` et une fonction **void** `tri3car(`**char** `*c1,` **char** `*c2,` **char** `*c3)`. On considère que l'ordre des caractères est l'ordre de leur code *ASCII*.
Solution:

```
void echange(char *c1, char *c2)
{
  char temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
}

void tri3car(char *c1,char *c2, char *c3)
{
  if (*c1 > *c2)
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
}
```

On suppose que ces fonctions appartiennent à un fichier `tri3car.c`.

Question 3. Que se passe-t-il lorsque l'on tente de compiler le code suivant (un fichier `main.c` à l'aide de la commande `gcc main.c tri3car.c tri3int.c`. Justifiez votre réponse et surtout expliquez comment il serait possible de remédier à ce problème.

```
#include <stdio.h>

void tri3int(int *, int *, int *);
void tri3car(char *, char *, char *);

int main()
{
  int i1 = 3, i2 = 2, i3 = 1;
  char c1 = 'x', c2 = 'z', c3 = 'y';
```

```
  tri3int(&i1, &i2, &i3);
  tri3car(&c1, &c2, &c3);

  printf("les entiers : %d, %d, %d\n", i1, i2, i3);
  printf("les caracteres : %c, %c, %c\n", c1, c2, c3);
}
```

Solution:
`static` devant les fonctions `echange`. Le qualificatif *static* rend les fonctions locales.

3 Le qualificatif `extern`

Maintenant on va se donner pour but de compter le nombre d'appels aux fonctions `echange`, `tri3car` et `tri3int`. Pour cela on va déclarer trois entiers dans le fichier `main.c` qui seront initialisés à 0.

Voici une première modification de `main.c`.

```
#include <stdio.h>

void tri3int(int *, int *, int *);
void tri3car(char *, char *, char *);

int c_echange = 0;
int c_tri3car = 0;
int c_tri3int = 0;

int main()
{
  int i1 = 3, i2 = 2, i3 = 1;
  char c1 = 'x', c2 = 'z', c3 = 'y';

  tri3int(&i1, &i2, &i3);
  tri3car(&c1, &c2, &c3);

  printf("les entiers : %d, %d, %d\n", i1, i2, i3);
  printf("les carateres : %c, %c, %c\n", c1, c2, c3);

  printf("nombre d'echanges : %d\n", c_echange);
  printf("nombre de tri3car : %d\n", c_tri3car);
  printf("nombre de tri3int : %d\n", c_tri3int);
}
```

Question 4. Comment accéder aux entiers `c_echange`, `c_tri3car` et `c_tri3car` depuis les autres fichiers ?
Solution:
Utilisation judicieuse de **`extern`**

Question 5. Modifiez `tri3int.c` et `tri3car.c` afin que `c_echange`, `c_tri3car` et `c_tri3car` comptent bien ce que l'on attend qu'ils comptent.

Solution:

```
extern int c_echange;
extern int c_tri3car;

static void echange(char *c1, char *c2)
{
  char temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
  c_echange++;
}

void tri3car(char *c1,char *c2, char *c3)
{
  if (*c1> *c2)
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
  c_tri3car++;
}
```

```
extern int c_echange;
extern int c_tri3int;

static void echange(int *c1, int *c2)
{
  int temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
  c_echange++;
}

void tri3int(int *c1, int *c2, int *c3)
{
  if (*c1> *c2)
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
  c_tri3int++;
}
```

Question 6. Le qualificatif **`extern`** peut-il être utilisé sur toutes déclarations d'une même variable (d'un même entier par exemple) ?

Solution:

Non, ce n'est pas possible, il y en a au moins une qui doit « définir » cette variable.

Question 7. Cela aurait-il du sens de déclarer une variable de la manière suivante ?

```
extern int ma_var = 0;
```

Pourquoi ?
Solution:
Non pas trop. Si la variable est **`extern`** c'est qu'elle est censée être définie et donc initialisée ailleurs.

4 Utilisation de `static` à l'intérieur des fonctions

Maintenant on change d'avis et on décide de ne compter que les appels aux fonctions `tri3car` et `tri3int`. Pour ce faire on souhaite déclarer des compteurs à l'intérieur même de ces fonctions. En plus on en modifie les types, elles deviennent désormais :
– **`int`** `tri3car(`**`char`** `*c1,`**`char`** `*c2,` **`char`** `*c3)` et
– **`int`** `tri3int(`**`int`** `*c1,` **`int`** `*c2,` **`int`** `*c3),`
et l'entier renvoyé est le nombre d'appels déjà effectués.

Question 8. Comment feriez vous ?
Solution:

```
#include <stdio.h>

int tri3int(int *, int *, int *);
int tri3car(char *, char *, char *);

int main()
{
  int i1 = 3, i2 = 2, i3 = 1;
  char c1 = 'x', c2 = 'z', c3 = 'y';

  printf("%d\n", tri3int(&i1, &i2, &i3));
  printf("%d\n", tri3car(&c1, &c2, &c3));
  //  printf("%d\n", tri3int(&i1, &i2, &i3));
  //  printf("%d\n", tri3car(&c1, &c2, &c3));

  printf("les entiers : %d, %d, %d\n", i1, i2, i3);
  printf("les caracteres : %c, %c, %c\n", c1, c2, c3);
}
```

```
static void echange(char *c1, char *c2)
{
  char temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
}

int tri3car(char *c1, char *c2, char *c3)
{
  if (*c1 > *c2)
```

```
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
  static int c_tri3car = 0;
  c_tri3car++;
  return c_tri3car;
}
```

```
static void echange(int *c1, int *c2)
{
  int temp;
  temp = *c1;
  *c1 = *c2;
  *c2 = temp;
}

int tri3int(int *c1, int *c2, int *c3)
{
  if (*c1 > *c2)
    echange(c1, c2);
  if (*c2 > *c3)
    echange(c2, c3);
  if (*c1 > *c2)
    echange(c1, c2);
  static int c_tri3int = 0;
  c_tri3int++;
  return c_tri3int;
}
```

5 Si on a le temps ...

Question 9. Qu'affiche le programme suivant ?

```
#include <stdio.h>

int f()
{
  static int cpt = 0;
  cpt++;
  return cpt;
}

int g()
{
  static int cpt = 1;
  cpt *= 2;
  return cpt;
}
```

```
int main()
{
  int i;
  for (i = 0; i < 3; i++)
    printf("%d\n", f());
  printf("--------\n");
  for (i = 0; i < 3; i++)
    printf("%d, %d\n", f(), g());
  return 0;
}
```

Solution:

```
1
2
3
--------
4, 2
5, 4
6, 8
```