Licence Informatique — **Initiation à la Programmation**
Université Bordeaux I — **Travaux Dirigés**

Pointeurs

- Simuler des programmes comportant des pointeurs.
- Comprendre le passage de paramètres en C.
- Comprendre la relation entre pointeurs et tableaux.

Exercice 1. Que produira l'exécution du programme suivant ?

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

void
main (void)
{
  int i, j;
  int *a, *b, **c;
  i = 1;
  j = 2;
  printf ("A␣␣i:␣%d,␣j:␣%d\n", i, j);
  a = &i;
  printf ("B␣␣*a:␣%d\n", *a);
  *a = *a + 5;
  printf ("C␣␣␣i:␣%d\n", i);
  i = i + 5;
  printf ("D␣␣*a:␣%d\n", *a);
  b = a;
  printf ("E␣␣*b:␣%d\n", *b);
  *b = *b + 5;
  printf ("F␣␣i:␣%d,␣*a:␣%d␣et␣*b:␣%d\n", i, *a, *b);
  b = &j;
  c = &b;
  printf ("G␣␣j:␣%d,␣*b:␣%d,␣**c:␣%d\n", j, *b, **c);
  printf ("H␣␣b:␣%p␣*b:␣%d␣␣c:␣%p␣*c:␣%p␣␣**c:␣%d\n", b, *b, c, *c, **c);
  exit (EXIT_SUCCESS);
}
```

Exercice 2. Que produira l'exécution du programme suivant ?

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

int
main (void)
{
  int tab[5] = { 2, 4, 6, 8, 10 };
  int *t = tab;
  size_t i;
  char mot[] = "arithmetique␣des␣pointeurs";
  char *s;

  for (i = 0; i < 5; i++)
    printf ("%d,␣%d␣\n", tab[i], *t++);

  for (s = mot; *s != '\0'; s++)
```

```
    printf ("%s\n", s);

  return EXIT_SUCCESS;
}
```

Exercice 3. Que produira l'exécution du programme suivant ?

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

void
echanger1 (int x, int y)
{
  int temp;
  temp = x;
  x = y;
  y = temp;
}

void
echanger2 (int *px, int *py)
{
  int temp;
  temp = *px;
  *px = *py;
  *py = temp;
}

int
main (void)
{
  int a = 1, b = 2;
  printf ("a=␣%d:␣b=␣%d\n", a, b);
  echanger1 (a, b);
  printf ("a=␣%d:␣b=␣%d\n", a, b);
  echanger2 (&a, &b);
  printf ("a=␣%d:␣b=␣%d\n", a, b);
  return EXIT_SUCCESS;
}
```

Exercice 4. Écrire les fonctions suivantes en adoptant le style tableau puis le style pointeur :

- `int strcmp (const char *s1, const char *s2);` qui retourne un entier négatif, nul, ou positif, si `s1` est respectivement inférieure, égale ou supérieure à `s2`.
- `char *strcpy (char *dest, const char *src);` qui recopie la chaîne `src` dans la chaine `dest`.
- `char *strstr (const char *meule_de_foin, const char *aiguille);`
  La fonction `strstr()` cherche la premiere occurence de la sous-chaîne `aiguille` au sein de la chaîne `meule_de_foin`. Les caractères `'\0'` de fin ne sont pas comparés. La fonction `strstr()` renvoie un pointeur sur le début de la sous-chaîne, ou `NULL` si celle-ci n'est pas trouvée.