4TIN303U - PROGRAMMATION C
Adresses et Pointeurs

▸ **Exercice 1.**

On considère le code source suivant. Expliquer le résultat de sa compilation et de son exécution.

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

int *f(int n) {
  int t[n];
  for (int i = 0; i < n; i++)
    t[i] = 1;
  return t;
}

void g(int v) {
}

int main(void) {
  int *tab = f(10);
  printf("%d\n", *tab);
  g(4);
  printf("%d\n", *tab);
  return EXIT_SUCCESS;
}
```

```
$ gcc -std=c99 -Wall erreur_src.c -o erreur_src
erreur_src.c: Dans la fonction <<f>>
erreur_src.c:8: attention : cette fonction retourne l'adresse d'une variable locale
$ ./error_src
1
-1079543952

```

ou :

```
error_src.c:8:10: warning: address of stack memory associated with local variable 't'
returned
[-Wreturn-stack-address]
return t;
^
1 warning generated.
$ ./error_src
1
1

```

▸ **Exercice 2.** Quel est l'effet de l'exécution du bloc d'instructions suivant ? Expliquer.

Vous dessinerez l'état de la mémoire au fil du programme.

```c
{
 int x = 1, y = 2, z[5];
 int *pi;

 pi = &x;
 *pi += y;
 y = *pi;
 *pi = 1;
 pi = z;
 *(pi + 2) = 5;
}
```

▸ **Exercice 3.** Que fait la fonction suivante ?

```c
bool f1(int *t, int n){
  for(int i=1 ; i<n ; i++)
    if ( *(t+i) > *(t+i-1) )
      return false;
  return true;
}
```

Réécrire la fonction en utilisant l'indexation (opérateur `[ ]`) et non pas des opérations arithmétiques sur les pointeurs.

▸ **Exercice 4.** Étudier le code de chacune des fonctions suivantes. Donner quelques lignes de code illustrant l'utilisation de ces fonctions. Expliquer les résultats obtenus.

```c
int g1(int *p, int i, int n) {
  assert(i >= 0 && i < n);
  return *(p+i);
}
```

```c
int* g2(int *p, int i, int n) {
  assert(i >= 0 && i < n);
  return (p+i);
}
```

```c
void g3(int *p, int n) {
  int *q = p;
  for (int i = 0; i< n; i++, q++)
    printf("%d\t%d\t%d\n", q[0], *q, *(p+i));
}
```

```c
void g4(char *s) {
  for ( ; *s != '\0'; s++)
    printf("%c\t%c\n", *s, s[0]);
}
```

▸ **Exercice 5.** Écrire un programme qui affiche la longueur d'une chaîne de caractères uniquement à l'aide de pointeurs.

▸ **Exercice 6.** Le programme suivant est il correct ? Si non, quels sont les erreurs ?

Si oui, qu'affiche-t-il à l'exécution ?

Vous devrez détailler l'état de la mémoire ligne par ligne.

```c
#include<stdio.h>
#include<stdlib.h>
#define N 5

int main(void) {
  int a = 1, b = 2, c = 3;
  int tab[N];
  for (int i = 0; i < N; i++){
    tab[i] = N-i;
  }
  *(tab + tab[a + *(tab + b)]) += *(tab + a);
  c -= tab[tab[*(tab + b)]];
  tab[*(tab + b - *(tab + 4)) - *(tab + c)] = tab[0];


  for (int i = 0; i < N; i++){
    printf("%d ",tab[i]);
  }
  printf("\n");
  return EXIT_SUCCESS;
}
```