

# TD 5 - Algorithmique et programmation Arbres

## Exercice 1

```
1  main()
2  {
3      int A = 1;
4      int B = 2;
5      int C = 3;
6      int *P1, *P2;
7      P1=&A;
8      P2=&C;
9      *P1=(*P2)++;
10     P1=P2;
11     P2=&B;
12     *P1-=*P2;
13     ++*P2;
14     *P1**=*P2;
15     A=++*P2**P1;
16     P1=&A;
17     *P2=*P1/=*P2;
18     return 0;
19 }
```

Donner pour chaque ligne les valeurs de A, B, C, P1 et P2 qui ont changé

## Exercice 2

Soit P un pointeur qui 'pointe' sur un tableau A :

```
1  int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
2  int *P;
3  P = A;
```

Quelles valeurs ou adresses fournissent ces expressions (lorsque c'est possible de savoir) :

1. \*P+2
2. \*(P+2)
3. &P+1
4. &A[4]-3
5. A+3
6. &A[7]-P
7. P+(\*P-10)
8. \*(P+\*(P+8)-A[7])

### Exercice 3

Ecrire de deux façons différentes, un programme qui vérifie sans utiliser une fonction de `<string>`, si une chaîne `maChaine` introduite au clavier est un palindrome :

1. en utilisant uniquement un tableau
2. en utilisant des pointeurs au lieu des indices numériques

### Exercice 4

Ecrire un programme qui lit deux matrices A et B de dimensions N et M respectivement M et P au clavier et qui effectue la multiplication des deux matrices.

Le résultat de la multiplication sera affecté à la matrice C, qui sera ensuite affichée. Utiliser le formalisme pointeur à chaque fois que cela est possible.