TD 2: Sémantique : environnement, mémoire et pile

Dans les quatre exercices ci-dessous, vous devez donner les *tableaux d'activation* des fonctions puis les états successifs de l'*environnement* et de la *mémoire* pour les principales étapes du programme et préciser les affichages à l'écran lorsqu'il y en a. On suppose que le segment de pile est constitué des adresses de 0 à 99 et que le segment de données statiques est constitué des adresses de 100 à 199.

1 – Affectations simples

```
1  #include <iostream>
2  using namespace std;
3  int main () {
4      int n1, n2;
5      cout << "Saisissez n1:" << endl;
6      cin >> n1;    // on suppose que l'on tape 2
7      cout << "Saisissez n2:" << endl;
8      cin >> n2;    // on suppose que l'on tape 3
9      n1 = n2;
10     n2 = n1;
11     cout << "n1 vaut " << n1 << endl;
12     cout << "n2 vaut " << n2 << endl;
13     return 0;
14 }
```

Modifiez ensuite ce code pour que les variables `n1` et `n2` échangent effectivement leur contenu, puis refaire les tableaux d'activation correspondant à cette nouvelle version du code.

2 – Première fonction

```
1  #include <iostream>
2  using namespace std;
3  int y;
4  int P1() {
5      int x = y * 3;
6      return x;
7  }
8  int main() {
9      int x;
10     x = 5;
11     y = 2 * x;
12     x = P1() + 3;
13     cout << "x = " << x << endl;
14     return 0;
15 }
```

3 – Plus compliqué

```
#include <iostream>
using namespace std;
int y = 10;
int P1() {
    int x;
    x = 7;
    return x + 1;
}
void P2() {
    int x, y;
    x = 3;
    y = P1();
    x = y;
}
void main() {
    int x;
    x = 1;
    y = P1() + 4;
    P2();
    cout << "x = " << x << endl;
}
```

4 – Avec des paramètres

```
#include <iostream>
using namespace std ;
int pgcd(int a, int b){
    int reste;
    do {
        reste = a % b;
        a = b;
        b = reste;
    } while (reste != 0);
    return a;
}
void main() {
    int a = 75, b = 60;
    int r = pgcd(a, b);
    cout << "pgcd(" << a << ", " << b << ")=" << r << endl;
}
```