

On définit le type `Pile` suivant :

```
class Pile {
    int sommet;
    int []valeurs;
    public Pile() {
        sommet = 0;
        valeurs = new int[100];
    }
    public int lire(int n) {
        return valeurs[sommet-n-1];
    }
    public void ecrire(int n,int valeur) {
        valeurs[sommet-n-1] = valeur;
    }
    public int depile() {
        return valeurs[--sommet];
    }
    public void empile(int valeur) {
        valeurs[sommet++] = valeur;
    }
}
```

Soit le programme suivant :

```
public static void main(String []a) {
    Pile pile = new Pile();
    boolean exit = false;
    int instruction;
    int registre;
    int [] memoire = new int[3];
    instruction = 1;
    while (!exit) {
        switch (instruction) {
            case 1:
                memoire[0] = 0;
                instruction++;
                break;
            case 2:
                memoire[1] = 0;
                instruction++;
                break;
            case 3:
                if (memoire[0]==10) instruction = 8;
                else instruction++;
                break;
            case 4:
                instruction++;
                pile.empile(instruction);
                pile.empile(memoire[0]);
                instruction = 9;
                break;
            case 5:
                memoire[1] += pile.depile();
                instruction++;
                break;
            case 6:
                memoire[0]++;
                instruction++;
                break;
        }
    }
}
```

```

        case 7:
            instruction = 3;
            break;
        case 8:
            exit = true;
            instruction++;
            break;
        case 9:
            pile.ecrire(0, 3*pile.lire(0));
            instruction++;
            break;
        case 10:
            registre = pile.depile();
            instruction = pile.lire(0);
            pile.ecrire(0, registre);
            break;
    }
}

```

1. Exécuter ce programme en notant soigneusement l'évolution des variables.
2. Que calcule ce programme ? Comment le fait-il ?
3. Écrire un programme naturel équivalent.
4. Faites l'opération inverse avec le programme suivant :

```

public static int f(int n) {
    int res;
    res=g(n);
    return res;
}
public static int g(int n) {
    int res;
    res=2+n;
    return res;
}
public static void main(String[] a) {
    int i;
    int tmp;
    tmp=1;
    for(i=0; i<10; i++){
        tmp*=f(i);
    }
    System.out.println(tmp);
}

```