

Afin de mieux visualiser le déroulement d'un programme et comprendre ses erreurs, il peut être utile de suivre l'évolution des variables, sur papier. Considérons l'extrait de programme suivant :

```
int temperature;
int nbJours = 7;
int nbPersonnes = 3;
int prixParJour = 15;
```

On peut déterminer l'état des variables à la fin de l'exécution de ce code :

temperature	nbJours	nbPersonnes	prixParJour
	7	3	15

La variable *temperature* est déclarée mais non initialisée : son contenu est indéfini, on ne le représente donc pas.

Exercice 1

À l'aide d'une feuille de brouillon, essayez de déterminer le résultat affiché par le programme suivant.

```
int nbPersonnes = 3;
int nbJours = 4;

int prixSejour = 0;
int prixJournee = 1;
repeat (nbJours)
{
    prixSejour = prixSejour + prixJournee;
    prixJournee = prixJournee + 2;
}
prixSejour = prixSejour + 14;

cout << "Prix par personne : " << prixSejour / nbPersonnes << endl;
```

Afficher/cacher la solution

Solution de l'exercice

Pour trouver le résultat affiché, on doit noter les valeurs successives que prennent les variables, afin de connaître leur valeur au moment de l'affichage. La méthode habituelle est de tracer des colonnes, une par variable, et d'écrire à chaque affectation la nouvelle valeur de la variable dans la colonne correspondante. Ci-dessous, on indique également, dans la première colonne, à quel endroit du programme correspond chaque ligne du tableau.

	nbPersonnes	nbJours	prixSejour	prixJournee
Avant la boucle	3	4	0	1
Fin du 1 ^{er} tour			1	3
Fin du 2 ^e tour			4	5
Fin du 3 ^e tour			9	7
Fin du 4 ^e tour			16	9
Fin			30	

Le résultat est donc :

↳ *Prix par personne : 10*

Dans la suite, essayez de faire cette petite analyse sur des programmes que vous rencontrez, avec un papier si nécessaire. Cela vous permettra de suivre de tête l'évolution des variables dans des programmes de plus en plus complexes quand vous aurez l'habitude.