Les variables
lecture / écriture

Pour écrire à l'écran

`cout` représente la fenêtre Terminal

affiche la *valeur* de la variable `n` (et non pas la lettre n)

ce qui est entre guillemets (") est affiché littéralement

les différents éléments à afficher doivent être séparés par le symbole `<<`

```
cout << "La variable n contient " << n << "." << endl;
```

affiche:

```
La variable n contient 4.
```

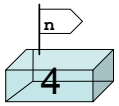
fait un "retour à la ligne": le prochain affichage se fera sur la ligne suivante de la fenêtre Terminal

Pour écrire à l'écran

On peut aussi utiliser `cout` pour afficher la valeur d'une expression:

```
cout << "Le double de " << n << " est " << 2 * n << "." << endl;
```

expression



Remarque

On peut écrire `cout` et `endl` simplement car le début du programme contenait la ligne:

```
using namespace std;
```

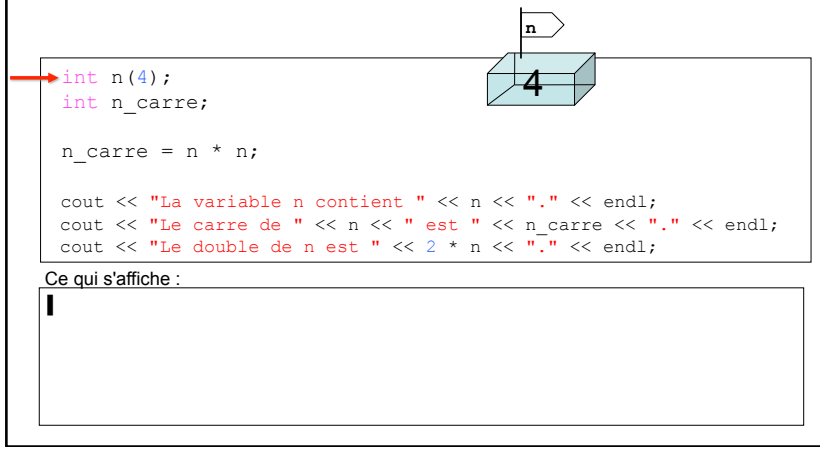
En l'absence de cette ligne, il faudrait écrire `std::cout` et `std::endl`.
Par exemple:

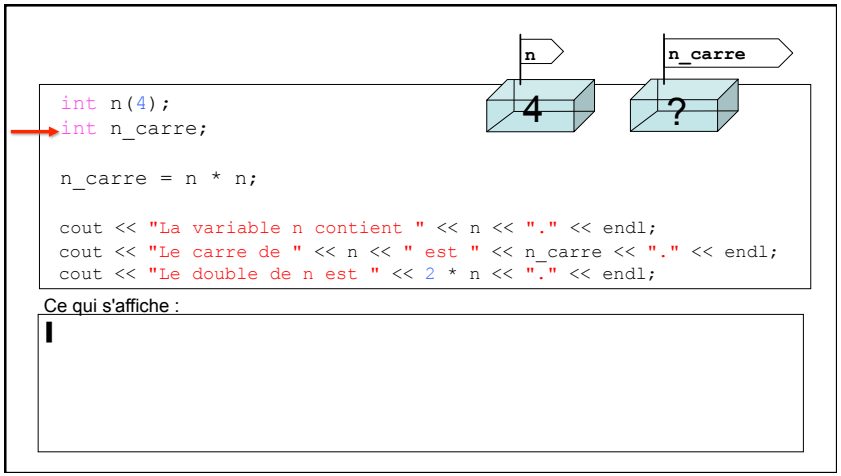
```
std::cout << "La variable n contient " << n << "." << std::endl;
```

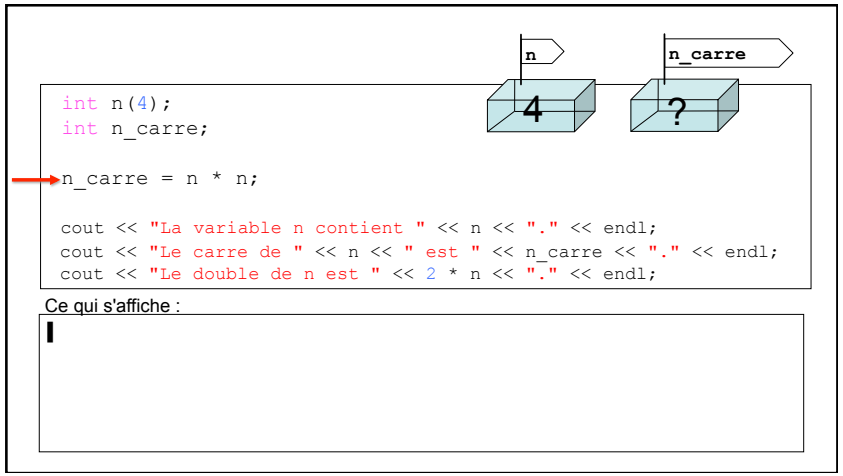
Déroulement du programme pas-à-pas

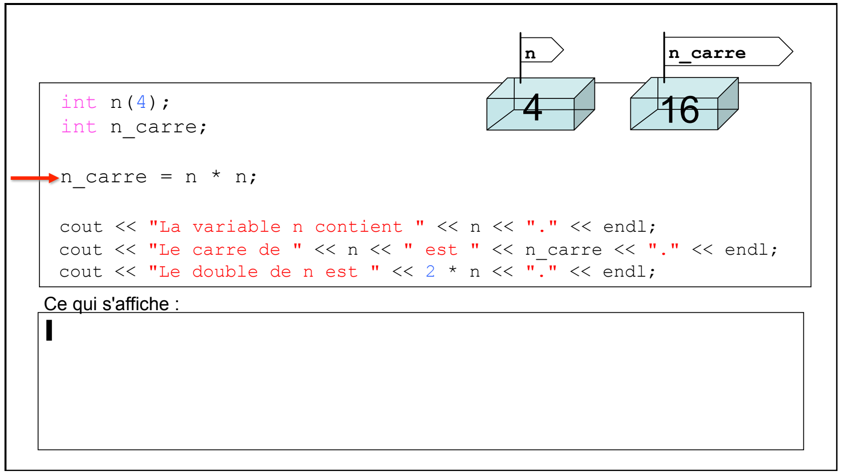
```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :





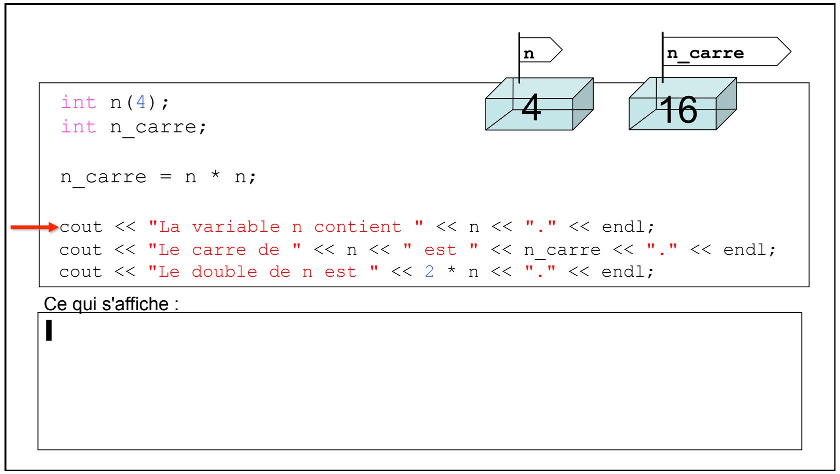




```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

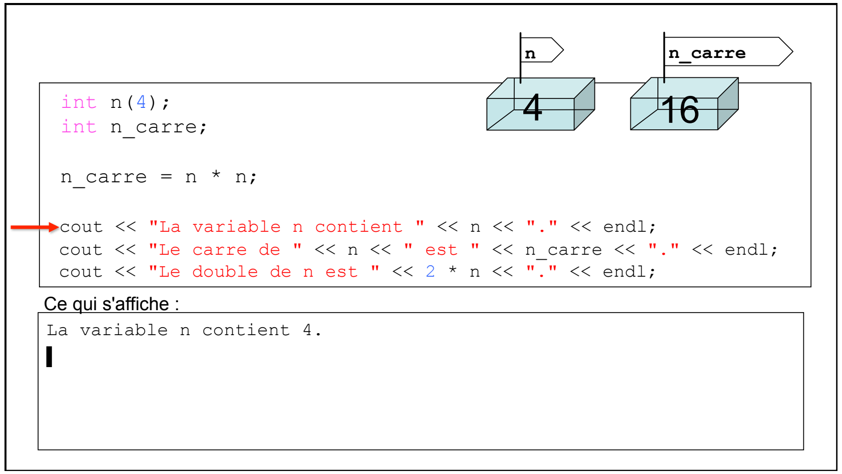
```
▌
```



```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

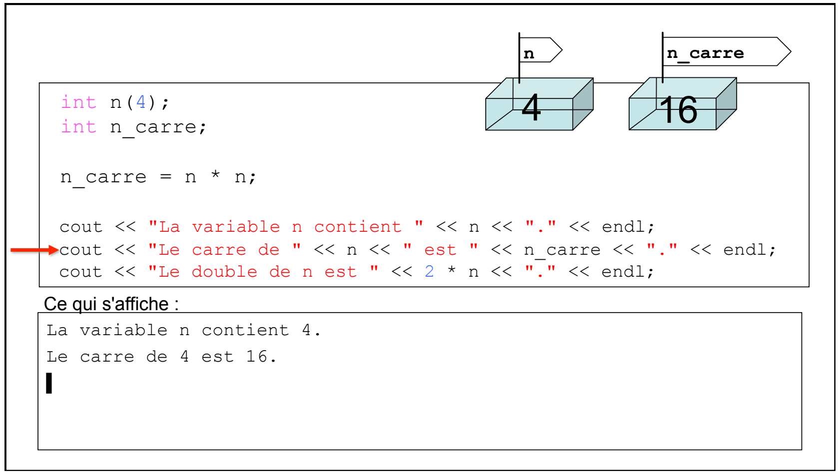
```
▌
```



```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

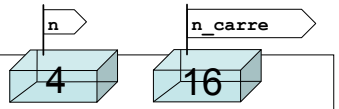
```
La variable n contient 4.
▌
```



```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

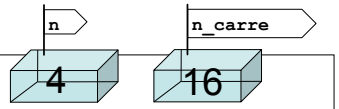
```
La variable n contient 4.
Le carre de 4 est 16.
▌
```




```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

```
La variable n contient 4.
Le carre de 4 est 16.
```




```
int n(4);
int n_carre;

n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
cout << "Le double de n est " << 2 * n << "." << endl;
```

Ce qui s'affiche :

```
La variable n contient 4.
Le carre de 4 est 16.
Le double de n est 8.
```

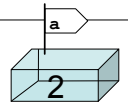
Qu'affiche ce programme ?

```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```

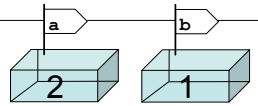
A: a, b
B: 1, 2
C: 2, 1
D: 2, 6

?

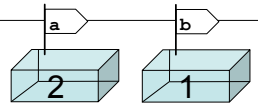
```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```



```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```

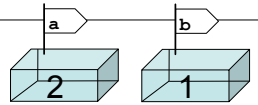



```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```



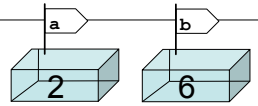

```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```

2 * (1 + 2) = 2 * 3 = 6

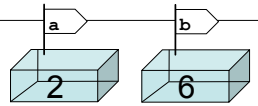



```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```

2 * (1 + 2) = 2 * 3 = 6



```
int a(2);
int b(1);

b = a * (b + 2);

cout << a << ", " << b << endl;
```



Affiche:
2, 6

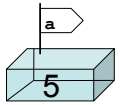
Qu'affiche ce programme ?

```
int a(5);
int b(a + 3);

a = 1;

cout << a << ", " << b << endl;
```

A: 5, 4
B: 1, 1
C: 1, 4
D: 1, 8

?

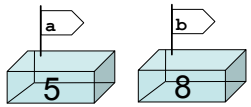
```
int a(5);
int b(a + 3);

a = 1;

cout << a << ", " << b << endl;
```



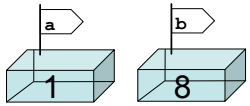

```
int a(5);
int b(a + 3);
```

5 + 3 = 8

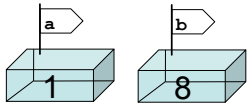
```
a = 1;

cout << a << ", " << b << endl;
```



```
int a(5);
int b(a + 3);

a = 1;

cout << a << ", " << b << endl;
```




```
int a(5);
int b(a + 3);

a = 1;

cout << a << ", " << b << endl;
```



Affiche:
`1, 8`

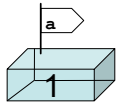
Qu'affiche ce programme ?

```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```

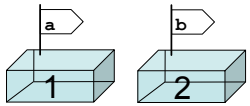
A: `1, 1`
B: `1, 2`
C: `2, 2`
D: `2, 1`

?

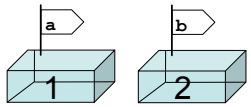
```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```



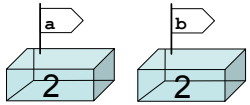
```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```



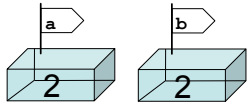

```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```



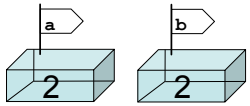

```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```




```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```



```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```



Affiche:

2, 2

Supposons qu'on ait déclaré et initialisé deux variables `a` et `b`.

Comment échanger leurs valeurs ?

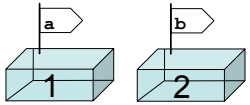
Les instructions:

```
a = b;
b = a;
```

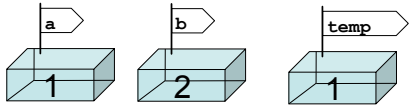
ne **marchent pas**, comme le montre la question précédente.

Solution: utiliser une variable intermédiaire:

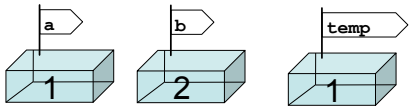
```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```



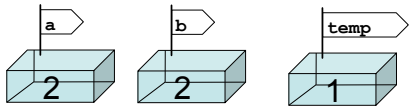

```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```



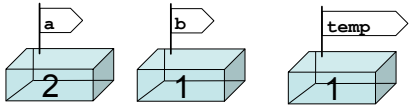

```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```




```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```




```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```




```
int a(1);
int b(2);
int temp(a);

a = b;
b = temp;
```

```
#include <iostream>
using namespace std;

int main()
{
  int n(4);
  int n_carre;

  n_carre = n * n;

  cout << "La variable n contient " << n << "." << endl;
  cout << "Le carre de " << n << " est " << n_carre << "." << endl;
  cout << "Le double de n est " << 2 * n << "." << endl;

  return 0;
}
```

```
#include <iostream>
using namespace std;

int main()
{
  int n;

  cout << "Entrez une valeur pour n:";
  cin >> n;

  int n_carre;

  n_carre = n * n;

  cout << "La variable n contient " << n << "." << endl;
  cout << "Le carre de " << n << " est " << n_carre << "." << endl;
  cout << "Le double de n est " << 2 * n << "." << endl;

  return 0;
}
```

Lire une valeur au clavier

`cin` représente le clavier

les différents éléments sont séparés par le symbole >>

nom de la variable dans laquelle sera stockée la valeur entrée au clavier

```
cin >> n;
```

Attention, uniquement des noms de variables peuvent figurer à droite du symbole >>

Par exemple, on ne peut pas faire:

~~`cin >> "Entrez un nombre" >> n;`~~

Il faut faire:

```
cout << "Entrez un nombre" << endl;
cin >> n;
```

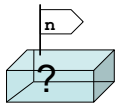
Par contre, on peut faire:

```
cin >> n1 >> n2 >> n3;
```

pour lire plusieurs valeurs à la suite.

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```

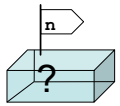


Ce qui s'affiche:

```
▌
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```

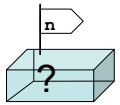


Ce qui s'affiche:

```
Entrez une valeur pour n:
▌
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```

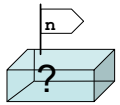


Ce qui s'affiche:

```
Entrez une valeur pour n:
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```



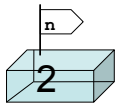
Ce qui s'affiche:

```
Entrez une valeur pour n:
2
```

supposons que l'utilisateur entre la valeur 2 au clavier (il faut presser *return* une fois la valeur tapée)

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```

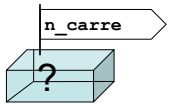


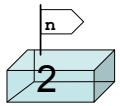
Ce qui s'affiche:

```
Entrez une valeur pour n:
2
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```



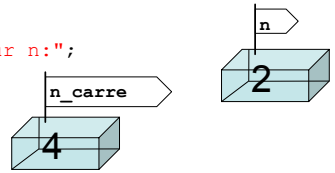


Ce qui s'affiche:

```
Entrez une valeur pour n:
2
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```

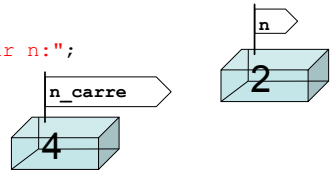


Ce qui s'affiche:

```
Entrez une valeur pour n:
2
```

Déroulement du programme pas-à-pas

```
int n;

cout << "Entrez une valeur pour n:";
cin >> n;

int n_carre;
n_carre = n * n;

cout << "La variable n contient " << n << "." << endl;
```



Ce qui s'affiche:

```
Entrez une valeur pour n:
2
La variable n contient 2.
```