

🔑 Générer un nombre aléatoire en C

Pour générer un nombre pseudo-aléatoire en C, on utilise la fonction `rand`. Celle-ci est codée dans la librairie standard, qu'il faut donc inclure. Cette fonction est un générateur, à chaque appel au cours d'une exécution elle donne un nouvel entier, la suite des entiers ainsi obtenus étant une suite de nombres pseudo-aléatoires. Cependant, utilisée telle quelle, cette fonction a le même comportement à chaque exécution. En commentant la ligne 6 du programme ci-contre avant compilation, on peut observer qu'il s'exécute toujours de la même manière.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<time.h>
4
5 int main(){
6     srand(time(NULL));
7     for(int i=0;i<=20;i++){
8         printf("%d\n", rand());
9     }
10    return 0;
11 }
```

Pour pallier ce problème, il faut initialiser cette suite avec une nouvelle valeur à chaque exécution. Pour cela, on utilise généralement la fonction `time` de la librairie `time.h` qui renvoie le nombre de secondes écoulées depuis le 01/01/1970 à 00h00mm00s.

Les entiers générés simulent une loi uniforme sur $\llbracket 0, N \rrbracket$ avec $N = \text{RAND_MAX}$, où RAND_MAX est une constante supérieure à $2^{15} - 1$ qu'on peut utiliser (et afficher par curiosité). Pour obtenir des entiers de $\llbracket 0, p - 1 \rrbracket$, on pourra prendre les valeurs obtenues modulo p , et pour obtenir des entiers de $\llbracket a, b \rrbracket$, on pourra prendre les valeurs obtenues modulo $p = b - a + 1$ puis leur ajouter a , même si ces opérations ne préservent pas tout à fait l'uniformité du tirage.

Exercice Écrire un programme qui affiche la proportion d'entiers plus petits que $\text{RAND_MAX}/2$ pour 1000 tirages effectués avec la fonction `rand`.