

GENERACIÓ DE

# NOMBRES ALEATORIS

# Generació de nombres aleatoris

## Nombres pseudo-aleatoris

- Molt sovint haurem de generar nombres aleatoris: llençar una moneda, llençar un dau, reordenar aleatòriament un conjunt de dades, generar nombres aleatoris d'una distribució, simular processos estocàstics...
- Com es generen els nombres aleatoris?
  - **Mètode 1:**
    - Mesurem algun fenòmen físic que és aleatori (o impossible de modelar) i els fem servir d'entrada a una funció que generarà nombres aleatoris.

# Generació de nombres aleatoris

## Nombres pseudo-aleatoris

- Molt sovint haurem de generar nombres aleatoris: llençar una moneda, llençar un dau, reordenar aleatòriament un conjunt de dades, generar nombres aleatoris d'una distribució, simular processos estocàstics...
- Com es generen els nombres aleatoris?
  - **Mètode 1:**
    - Mesurem algun fenòmen físic que és aleatori (o impossible de modelar) i els fem servir d'entrada a una funció que generarà nombres aleatoris.



# Generació de nombres aleatoris

## Nombres pseudo-aleatoris

- Molt sovint haurem de generar nombres aleatoris: llençar una moneda, llençar un dau, reordenar aleatòriament un conjunt de dades, generar nombres aleatoris d'una distribució, simular processos estocàstics...
- Com es generen els nombres aleatoris?
  - **Mètode 1:**
    - Mesurem algun fenòmen físic que és aleatori (o impossible de modelar) i els fem servir d'entrada a una funció que generarà nombres aleatoris.
  - **Mètode 2:**
    - Fer servir algorismes computacionals que són capaços de produir seqüències molt llargues de nombres aparentment aleatoris, però que en el fons estan determinats per un valor inicial concret (la llavor o seed).
    - Per tant, tota la seqüència de nombres aleatoris es pot reproduir si coneixem la llavor.
    - Això genera nombres pseudo-aleatoris.
    - La llavor acostuma a ser l'hora del sistema.



# Generació de nombres aleatoris

## Llibreria `<stdlib.h>`

- Per generar nombres pseudo-aleatoris en C, farem servir la funció **`srand()`**, disponible a la llibreria **`stdlib.h`**

```
int rand(void);
```

Retorna un valor pseudo-aleatori enter entre `0` i `RAND_MAX` (0 i `RAND_MAX` inclosos).

`RAND_MAX`

Valor enter constant que depèn de la implementació. Es garanteix que aquest valor serà almenys `32767`.

```
void srand (unsigned seed);
```

`srand()` configura la llavor (seed) del generador de nombres aleatoris que utilitza `rand()`.

**Només s'ha d'executar un cop**, al principi del programa.

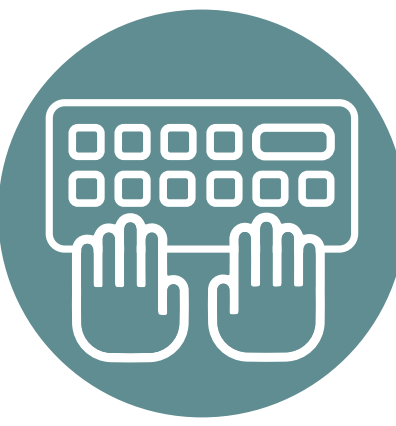
Si `rand()` és cridat abans de cridar `srand()`, `rand()` es comporta com si tingués llavor 1: `srand(1)`.

Si la llavor és constant (e.g. `srand(123)`) la seqüència de valors aleatoris sempre serà la mateixa en execucions diferents

El més comú és fer servir l'hora del sistema com a llavor. `srand(time(NULL));`

Per fer-ho, necessitem cridar la funció **`time()`**, que està disponible a la llibreria **`<time.h>`**

# Generació de nombres aleatoris



## Exemple

- Com es genera un nombre aleatori.
- Escriviu aquest programa i comproveu que dona valors aleatoris.
- Què passa si l'executeu molts cops seguits sense que passi massa temps entre ells?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     srand(time(NULL)); // use current time as seed for random generator
8     int aleatori = rand();
9     printf("Valor aleatori value en rang [0,%d]: %d\n", RAND_MAX, aleatori);
10 }
```

# Generació de nombres aleatoris

## Exemple

- Com es genera un valor aleatori entre  $[0, 1]$  ?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     srand(time(NULL)); // use current time as seed for random generator
8
9     int aleatori;
10    float aleatori_0_1;
11
12    aleatori = rand();
13    aleatori_0_1 = (float) aleatori / RAND_MAX;
14
15    printf("El nombre aleatori en rang [0,1] es %.4f\n", aleatori_0_1);
16 }
```

# Generació de nombres aleatoris

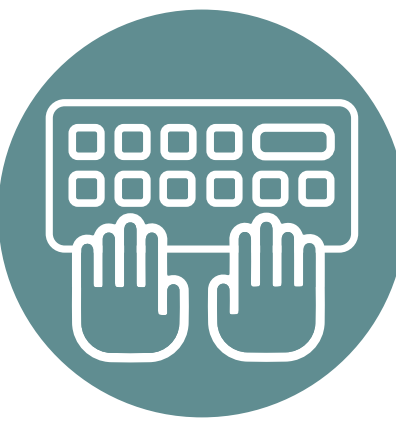
## Exemple

- Com es genera un valor aleatori entre [MIN, MAX] ?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 #define MIN 1
6 #define MAX 6
7
8 int main(void)
9 {
10     srand(time(NULL)); // use current time as seed for random generator
11
12     int aleatori;
13     float aleatori_0_1;
14     int aleatori_min_max;
15
16     aleatori = rand();
17     aleatori_0_1 = (float) aleatori / RAND_MAX;
18     aleatori_min_max = aleatori_0_1 * (MAX - MIN + 1) + MIN;
19
20     printf("El nombre aleatori en rang [%d,%d] es %d\n", MIN, MAX, aleatori_min_max);
21 }
```



# Generació de nombres aleatoris



## Exemple

- Com es genera un valor aleatori **enter** entre [MIN, MAX] ?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 /* Exemple per llençar un dau amb el modul */
6 #define MOD 6
7
8 int main(void)
9 {
10     srand(time(NULL)); // use current time as seed for random generator
11
12     int aleatori;
13     aleatori = rand() % MOD + 1;
14     printf("He tirat un dau i ha sortit %d\n", aleatori);
15
16 }
```

# Generació de nombres aleatoris

## Llavor fixa

- Generalment, volem que els nombres aleatoris siguin diferents cada cop que executem el programa.

Per exemple, executant aquest codi  
5 vegades, obtenim:

```
$ ./aleatoris
5 2 2 1 5 1 1 5 3 6

$ ./aleatoris
1 4 4 4 6 5 1 2 5 6

$ ./aleatoris
2 5 2 5 3 6 1 3 4 6

$ ./aleatoris
4 6 5 2 4 4 1 6 6 6

$ ./aleatoris
6 2 1 5 5 2 1 2 3 1
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MOD 6
#define TIRADES 10

int main()
{
    int aleatori;
    srand(time(NULL)); // Llavor: temps

    for(int i=0; i<TIRADES; i++){
        aleatori = rand() % MOD + 1;
        printf("%d \n", aleatori);
    }
    return 0;
}
```

# Generació de nombres aleatoris

## Llavor fixa

- Generalment, volem que els nombres aleatoris siguin diferents cada cop que executem el programa.
- Ara bé, quan estem programant, a vegades resulta útil fixar la seqüència, perquè sempre que executis el programa es generi la mateixa seqüència de nombres aleatoris i així poder detectar i corregir errors.
- Per fer-ho, el que farem és que la llavor, en comptes de ser l'hora del sistema (diferent cada cop) la fixarem a un nombre qualsevol, que serà sempre igual:

I ara, si executem  
el codi 5  
vegades,  
obtenim:

```
$ ./aleatoris
5 5 1 2 2 3 3 6 5 4

$ ./aleatoris
5 5 1 2 2 3 3 6 5 4

$ ./aleatoris
5 5 1 2 2 3 3 6 5 4

$ ./aleatoris
5 5 1 2 2 3 3 6 5 4

$ ./aleatoris
5 5 1 2 2 3 3 6 5 4
```

```
int aleatori;
srand(1234); // Llavor: fixada

for(int i=0; i<TIRADES; i++){
    aleatori = rand() % MOD + 1;
    printf("%d \n", aleatori);
}
```