

# Approfondimento : numeri pseudocasuali

Il funzionamento di un computer è un processo deterministico

- Regolato da leggi rigide
- Completamente determinato dalle condizioni iniziali

Questo comporta che è molto difficile ottenere vera casualità (ossia eventi statisticamente non predicibili) con un computer.

# Pseudocasualità

Un processo pseudocasuale è un processo che sembra essere casuale ma non lo è.

Una sequenza pseudocasuale sembra possedere **casualità statistica** anche se viene generato da un processo interamente deterministico.

Tale processo è più facile da produrre che uno completamente casuale. Inoltre può essere usato per ricreare molte volte la stessa sequenza di numeri (utile talvolta per testare il software).

# Generatori di numeri pseudocasuali

Un generatore di numeri pseudocasuali è un algoritmo per generare sequenze di numeri che approssimino le proprietà statistiche di una vera sequenza casuale.

La sequenza non è veramente casuale ma completamente determinata da un piccolo insieme di valori iniziali e da un singolo valore davvero casuale chiamato **seme** della sequenza.

- Usando due volte lo stesso seme si ottiene la stessa sequenza di numeri!

# Numeri pseudocasuali in C

In C abbiamo un generatore di numeri pseudocasuali.

- per usarlo dobbiamo includere la libreria `stdlib.h`

```
#include<stdio.h>
#include<stdlib.h>
int main() {
    int numero=0;
    srand(42); /* inicializzo il seme */
    numero = rand();
```

La funzione `rand()` restituisce un numero (ogni volta diverso) compreso fra 0 e `RAND_MAX` (una costante molto grande, di solito  $2^{31}$  ma comunque più grande di  $2^{15}$ ).

# Numeri pseudocasuali in C

```
#include<stdio.h>
#include<stdlib.h>
int main() {
    int numero=0;
    srand(42); /* inicializzo il seme */
    numero = rand();
}
```

La funzione `srand` usa il numero passato come **seme** per inizializzare il generatore di numeri casuali

- se usate lo stesso seme ottenete la stessa sequenza di numeri

Come ottenere un seme veramente casuale?

Come ottenere numeri in un determinato range?

# Inizializzazione del seme

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int numero=0;
    srand( time(NULL) ); /* inizializzo il seme */
    numero = rand();
}
```

La funzione `time()` (con argomento `NULL`) restituisce il tempo attuale in secondi trascorsi dall'Epoca (la mezzanotte del 1/1/1970).

- Dobbiamo includere la libreria `time.h`

Ogni volta che eseguite il programma ottenete un seme diverso (purchè non eseguito troppo rapidamente rispetto all'esecuzione precedente).

# Numero casuale in un range

La funzione `rand()` restituisce numeri in un range molto grande (tipicamente fra 0 e  $2^{31}$ ).

Normalmente siamo interessati a range molto più piccoli e (talvolta) che non iniziano da 0.

Usiamo l'operazione di modulo `%` per restringere il range:

- `rand() % 100`  
mappa il numero ottenuto nel range **[0 , 99]**

e aggiungiamo un eventuale offset:

- `(rand() % 100) + 50`  
mappa il numero nel range **[50 , 149]**

# Tutto insieme

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int numero=0;
    srand( time(NULL) ); /* inizializzo il seme */
    numero = (rand() % 100) + 50;
```

Con questo codice, quindi, potete ottenere un numero casuale compreso fra 150 e 149.

Notate che `srand()` va usato una sola volta, prima del primo utilizzo di `rand()`.

Inoltre potete ottenere numeri casuali in range diversi variando ogni volta i valori del modulo e dell'offset.

# Esempio

Scrivere un programma che chiede all'utente ampiezza e offset di un range e poi stampa 3 numeri casuali in quel range.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int numero=0, ampiezza, offset;
    srand( time(NULL) ); /* inicializzo il seme */
    printf("Inserisci l'ampiezza:");
    scanf("%d", &ampiezza);
    printf("Inserisci l'offset:");
    scanf("%d", &offset);
    numero = (rand() % ampiezza) + offset;
    printf("primo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("secondo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("terzo numero: %d\n", numero);
}
```

# Esempio

Scrivere un programma che chiede all'utente ampiezza e offset di un range e poi stampa 3 numeri casuali in quel range.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int numero=0, ampiezza, offset;
    srand( time(NULL) ); /* inizializzo il seme */
    printf("Inserisci l'ampiezza:");
    scanf("%d", &ampiezza);
    printf("Inserisci l'offset:");
    scanf("%d", &offset);
    numero = (rand() % ampiezza) + offset;
    printf("primo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("secondo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("terzo numero: %d\n", numero);
}
```

# Esempio

Scrivere un programma che chiede all'utente ampiezza e offset di un range e poi stampa 3 numeri casuali in quel range.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int numero=0, ampiezza, offset;
    srand( time(NULL) ); /* inicializzo il seme */
    printf("Inserisci l'ampiezza:");
    scanf("%d", &ampiezza);
    printf("Inserisci l'offset:");
    scanf("%d", &offset);
    numero = (rand() % ampiezza) + offset;
    printf("primo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("secondo numero: %d\n", numero);
    numero = (rand() % ampiezza) + offset;
    printf("terzo numero: %d\n", numero);
}
```