

Generazione di numeri pseudo-casuali

Perchè è difficile?

- Il computer esegue deterministacamente le istruzioni fornite dal programma
 - Questo comporta che è molto difficile ottenere vera casualità (**ossia eventi statisticamente non predicibili**) con un computer
- Il C fornisce delle funzioni di libreria generano sequenze di numeri pseudo-casuali
 - Vedremo **rand()** e **srand()**

Sequenze pseudo-casuali

$$x_0 \ x_1 \ x_2 \ \dots \ x_{k-1} x_k \ \dots$$

- Sono sequenze in cui **ciascun elemento (x_k)** è costruito a partire dall'elemento precedente (x_{k-1}) usando vari algoritmi
 - Sono costruiti in modo che la distribuzione dei valori ottenuti approssimi la distribuzione uniforme in un certo intervallo
- Il primo elemento (x_0) è detto **seme** della sequenza
- A partire dallo stessa seme la sequenza generata è sempre la stessa

Generare la sequenza in C

$$x_0 \ x_1 \ x_2 \ \dots \ x_{k-1} x_k \ \dots$$

- Si usano due funzioni di libreria standard **rand()** e **srand()**

Generare la sequenza in C

$$\boxed{x_0} x_1 x_2 \dots x_{k-1} x_k \dots$$

- Si usano due funzioni di libreria standard **srand()** e **rand()**
- **srand()** viene utilizzata una sola volta per fissare il seme

Generare la sequenza in C

$$x_0 \quad x_1 \quad x_2 \quad \dots \quad x_{k-1} \quad x_k \quad \dots$$

- Si usano due funzioni di libreria standard **srand()** e **rand()**
- **srand()** viene utilizzata una sola volta per fissare il seme
- **rand()** viene utilizzata per generare il resto della sequenza (ogni volta che viene chiamata fornisce il valore successivo)

Vediamo un esempio

- Generiamo i primi due numeri della sequenza di seme 42

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int n1, n2;
    srand(42);
    n1 = rand();
    n2 = rand();
    printf("Primi due numeri: %d e %d", n1, n2);
    return 0;
}
```

Vediamo un esempio

- Generiamo i primi due numeri della sequenza di seme 42

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

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

```
int main(void) {
```

```
    int n1, n2;
```

```
    srand(42);
```

```
    n1 = rand();
```

```
    n2 = rand();
```

```
    printf("Primi due numeri: %d e %d", n1, n2);
```

```
    return 0;
```

```
}
```

Serve a usare le funzioni
rand() e srand()

Vediamo un esempio

- Generiamo i primi due numeri della sequenza di seme 42


```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int n1, n2;
    srand(42);
    n1 = rand();
    n2 = rand();
    printf("Primi due numeri: %d e %d", n1, n2);
    return 0;
}
```

Fissa il seme al valore 42

Vediamo un esempio

- Generiamo i primi due numeri della sequenza di seme 42

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int n1, n2;
    srand(42);
    n1 = rand();
    n2 = rand();
    printf("Primi due numeri: %d e %d", n1, n2);
    return 0;
}
```




Genera x_1

Vediamo un esempio

- Generiamo i primi due numeri della sequenza di seme 42

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int n1, n2;
    srand(42);
    n1 = rand();
    n2 = rand();
    printf("Primi due numeri: %d e %d", n1, n2);
    return 0;
}
```



Genera x_2

Intervallo di generazione

- 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})
- Si può stabilire sperimentalmente Come ?

Intervallo di generazione

- 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})
- Si può stabilire sperimentalmente

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

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

```
int main(void) {
```

```
    printf ("RAND_MAX: %d\n", RAND_MAX);
```

```
    return 0;
```

```
}
```

Cambiare il seed/ seme iniziale

- Si può utilizzare la funzione **time ()** che restituisce il numero di secondo passati dal 1 Gennaio 1970 per fornire il seme
- Se le esecuzioni non sono troppo veloci (avvengono in secondi diversi) abbiamo un valore diverso per il seme assicurato
- Vediamo un esempio

Esempio

- Generare una sequenza di N interi nell'intervallo `[0, RAND_MAX]` con seme generato usando `time()`

Esempio

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 100

int main(void) {
    int i;
    srand(time(NULL));
    for (i=0; i< N; i++) {
        printf ("%d\n", rand());
    }
    return 0;
}
```


Generare in intervalli diversi

- Intervallo intero $[A,B]$?

Generare in intervalli diversi

- Intervallo intero [A,B]

$$r = A + rand() \% (B - A + 1)$$

Esempio

- Generare una sequenza di N interi in $[A, B]$

```
int main(void) {
    int i, r;
    srand(time(NULL));
    for (i=0; i< N; i++) {
        r = A + rand() % (B-A + 1);
        printf ("%d\n", r);
    }
    return 0;
}
```

Generare in intervalli diversi

- Intervallo intero [A,B]

$$r = A + rand() \% (B - A + 1)$$

- Intervallo reale [0,1)

$$r = rand() / (RAND_MAX + 1.0)$$

- Intervallo reale [a,b) generico ?