COESIONE ITALIA 21-27 PIEMONTE
Cofinanziato dall'Unione europea
REGIONE PIEMONTE

WEB DEVELOPER
Fondamenti di Programmazione
Massimo PAPA

Generatore di numeri casuali in C++

Intervento realizzato da ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Generatore di numeri casuali in c++

`rand()`

E’ la funzione che viene utilizzata per generare un numero casuale compreso tra **0** e **RAND_MAX**.

dove RAND_MAX è un valore che cambia a seconda del compilatore usato (in genere 32767).

Generatore di numeri casuali in c++

```
#include<iostream>
#include<cstdlib>
using namespace std;
int main()
{
   cout<<rand()%1000<<endl;
}
```

Il codice a fianco mostra come generare un numero compreso tra 0 e 999

ATTENZIONE: I numeri generati non sono veramente casuali (*numeri pseudo-casuali*)

Generatore di numeri casuali in c++

L'istruzione `rand()%1000` genera un numero intero compreso tra 0 e 999.

Se utilizziamo l'istruzione `rand()%N`, infatti, otterremo un numero compresi tra 0 ed N-1.

Per impostare un massimo si ricorre all'operatore modulo (%) che, serve per calcolare il resto di una divisione intera e questo spiega perché il valore massimo può essere N-1.

Generatore di numeri casuali in c++

```
rand()%5
```

Facciamo un esempio: un qualsiasi numero diviso per 5 può dare come resto 0, 1, 2, 3 e 4... non può dare 5 perché altrimenti avremmo una divisione intera senza resto!

.

Esempi:

```
rand()%2 // genera un numero compreso tra 0 e 1

rand()%100 // genera un numero compreso tra 0 e 99

rand()%43 // genera un numero compreso tra 0 e 42

rand()%10+5 // genera un numero compreso tra 5 e 14
```

Nell'ultima istruzione abbiamo aggiunto +5 (questo numero prende il nome di *offset*) a entrambi i numeri dell'intervallo (il minore ed il maggiore).

I numeri casuali sono sempre gli stessi!

Se lanciamo i programmi visti otteniamo sempre gli stessi numeri.

Questo accade in quanto la funzione `rand()` non genera un numero realmente casuale ma restituisce il prossimo valore pseudo-casuale nella sequenza di valori che va da 0 a **`RAND_MAX`**. Per ottenere numeri differenti è necessario cambiare il punto iniziale (**seme**) di quella sequenza usando `srand()`.

L’istruzione srand()

Se lanciamo i programmi visti otteniamo sempre gli stessi numeri.

Questo accade in quanto la funzione **`rand()`** non genera un numero realmente casuale ma restituisce il prossimo valore pseudo-casuale nella sequenza di valori che va da 0 a **`RAND_MAX`**. Per ottenere numeri differenti è necessario cambiare il punto iniziale (**seme**) di quella sequenza usando **`srand()`**.

L’istruzione srand()

Esaminiamo il seguente programma che genera 10 numeri casuali

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


int main()
{
  // inizializzo il generatore
di numeri pseudo-casuali
  srand(time(NULL));


  for(int i=0;i<10;i++) {
    // genero un numero casuale
compreso tra 0 e 9
    cout<<rand()%10<<endl;
  }
}
```

L’istruzione srand() e time()

La funzione ***`srand()`*** serve a inizializzare la funzione per la generazione dei numeri casuali: senza di essa allo stesso *seed* (seme) il programma estrarrebbe sempre gli stessi numeri casuali. Per rendere casuale il seme è quello di impostarlo con ***`time(NULL)`*** o ***`time(0)`*** incorporando la relativa funzione che si trova nella libreria ***`ctime`*** (che abbiamo incluso prima).

L’istruzione srand() e time()

Si noti che nel primo esempio mostrato prima la funzione **`srand()`** non era presente: tutte le volte che **`rand()`** viene usata senza che prima non si sia richiamata **`srand()`** è come se si richiamasse **`srand(1)`**... questo è il motivo per cui nell'esempio precedente si stampava sempre lo stesso numero.