

Sperimentazioni di Fisica I

mod. A – Laboratorio 6

Generazione di Numeri Casuali

*Dipartimento di Fisica e Astronomia “G. Galilei”,
Università degli Studi di Padova*

Generazione di numeri casuali in C++

- Prima dello standard C++11, i programmi in C++ utilizzavano la **funzione** `rand()` del C, oppure si appoggiavano a **librerie esterne** (per esempio le librerie **BOOST**)
- con il nuovo standard sono state portate nelle librerie del C++ un insieme di classi nuove (header file: `<random>`):
 - generatori di numeri casuali
 - classi per generare da distribuzioni di probabilità definite

```
#include <iostream>
#include <random>
using namespace std;
```

```
int main()
{
    default_random_engine e;

    for (size_t i=0; i<10; i++) {
        cout << e() << " ";
    }
    return 0;
}
```

```
$ g++ -std=c++11 rnd1.cxx
```

```
$ ./a.out
```

```
16807 282475249 1622650073 984943658 1144108930 ...
```

```
cout << e.min() << " <-> " << e.max();
```

```
$ 1 <-> 2147483646
```

Generazione di numeri interi

- È possibile generare dei numeri da una distribuzione specifica, in un range di valori fissato dall'utente
- si definisce un **generatore di numeri casuali**
- e una **distribuzione di probabilità** secondo la quale generare

```
#include <iostream>
#include <random>
using namespace std;
int main()
{
    uniform_int_distribution<unsigned> u(0, 9);
    default_random_engine e;

    for (size_t i=0; i<10; i++) {
        cout << u(e) << " ";
    }
    cout << endl;
}
```

```
$ g++ -std=c++11 rnd1.cxx
$ ./a.out
0 1 7 4 5 2 0 6 6 9
$ ./a.out
0 1 7 4 5 2 0 6 6 9
```

Generazione di numeri reali

- Spesso è necessario generare dei numeri reali, per esempio da una distribuzione uniforme
- è sufficiente cambiare il tipo della distribuzione e specificare gli estremi del dominio di generazione

```
#include <iostream>
#include <random>
using namespace std;
int main()
{
    uniform_real_distribution<unsigned> u(0,1);
    default_random_engine e;

    for (size_t i=0; i<10; i++) {
        cout << u(e) << " ";
    }
    cout << endl;
}
```

```
$ g++ -std=c++11 rnd1.cxx
$ ./a.out
0.131538 0.45865 0.218959 0.678865 0.934693 0.519416 ...
```

Generazione di numeri reali non uniformi

- una distribuzione di probabilità che useremo è quella **gaussiana**
- generiamo da una distribuzione con **valor medio** ($\mu = 4$) e **deviazione standard** ($\sigma = 1.5$)

```

default_random_engine e;
normal_distribution<> norm(4, 1.5);

vector<unsigned> hist(9);

for (size_t i = 0; i < 100; ++i) {

    auto x = lround(norm(e)); // arrotonda all'intero piu vicino

    if (x < hist.size())
        ++hist.at(x);
}

for (size_t i=0; i<hist.size(); i++)
    cout << i << ": "
         << string(hist[i], '*')
         << endl;

```

```

$ g++ -std=c++11 norm.cxx
$ ./a.out
0: **
1: ***
2: *****
3: *****
4: *****
5: *****
6: *****
7: ***
8:

```

Comandi per Compilare

Comando generale:

```
g++
```

Per utilizzare le features dello standard C++11 :

- dalla versione GCC 4.7 in poi:

```
g++ -std=c++11
```

- per le versioni precedenti:

```
g++ -std=c++0x
```