Termine iscrizioni: 6/11/25

Corso di sicurezza: entro 17/11

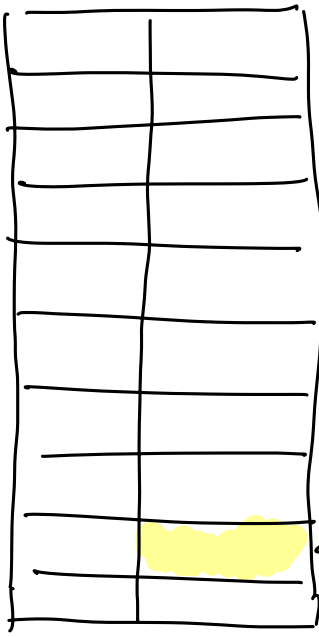
sito web corso: www.rahatlou.net

studenti

array statici in C

```
double dati[100];
int    studenti[160][2];
```

studenti[0][1]



colonna 0: matricola

colonna 1: date di nascita

20020221

```
studenti[0][0] = 2211345;
studenti[0][1] = 20020221;
```

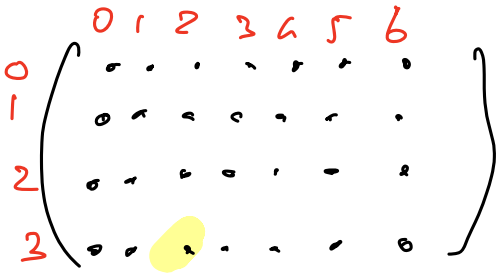
```
#define NSTUDENTI 160
#define MATRICOLA 0
#define DOB       1

int main() {
    int studenti[NSTUDENTI][2];
    studenti[0][MATRICOLA] = 2121332;
    studenti[0][DOB] = 20060606;
```

```
int presenze[NSTUDENTI] = {0};
int npresenti = 101;
for(int i=0; i< npresenti; i++) {
    presenze[i] = 2221347;
    printf("matricola studente %d:", i+1);
    scanf("%d", &presenze[i]);
}
```

usuale per tutti elementi array.

```
double mat[4][7];
scanf("%lf", &mat[3][2]);
```

indice riga = 3




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

#define NMAX 10

int main() {
  // array statico di lunghezza NMAX
  double vettore[NMAX] = {0};
  int i,j, n;

  // numero effettivo di valori da trattare n < NMAX
  do {
    printf("numero massimo di dati da inserire (< %d): ", NMAX);
    scanf("%d", &n);
  } while(n<=0 || n > NMAX);


  printf("ora inserisci %d elementi del vettore\n", n);
  for(i = 0; i < n; i++) {
    vettore[i] = i + 1;
    printf("vettore[%d]: ",i);
    scanf("%lf", &vettore[i] );
  } // ciclo input

  // ora stampa i valori nel vettore
  printf("ecco i valori salvati:\n");
  for(i = 0; i < n; i++) {
    printf("vettore[%d] = %.3f\n",i, vettore[i]);
  } // ciclo output

} // main
```

n: lunghezza effettive di elementi da usare.

vettore[0]: 13 prompt sulla riga di comando

```
printf("vettore - .\n", i);
```

vettore[0]:
13 prompt

```
shamacmini:material rahatlou$ gcc -o /tmp/app array1.c
shamacmini:material rahatlou$ /tmp/app
numero massimo di dati da inserire (< 10): 3
ora inserisci 3 elementi del vettore
vettore[0]: -1223.342
vettore[1]: 1.23e5
vettore[2]: 2232.323
ecco i valori salvati:
vettore[0] = -1223.342
vettore[1] = 123000.000
vettore[2] = 2232.323
```

1.23×10^5

Generazione numeri casuali:

Standard library: `#include <stdlib.h>`

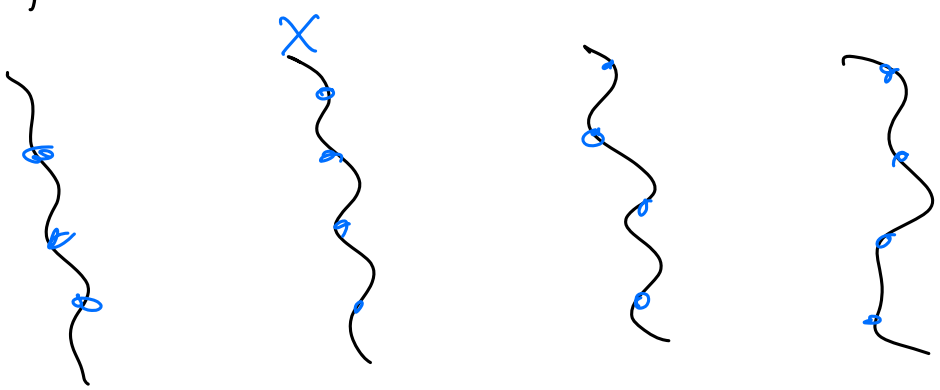
```
#include <stdlib.h>
#include <time.h>

int main() {

    srand48( time(0) );
    // Inizializza sequenza num. casuali

    int n,m;
    n = lrand48();
// num. casuale intero ∈ [0, RAND_MAX]
```

1 volta per tutto il programma immediatamente inizio di main()

RAND_MAX: intero più grande disponibile

```
printf("RAND_MAX = %d", RAND_MAX);
```

sequenza di num. pseudo-casuali:



Dado: 1, 2, 3, 4, 5, 6

```
int n = 123479123;
```

$\frac{n}{m}$ resto $\in [0, \dots, m-1]$ resto $\frac{n}{m} \equiv n \,\%\, m$

Dado: $n = (\text{rand}()\,\%\,6) + 1$

$\underbrace{}_{0,1,2,3,4,5}$

$n \in [1, 6]$

Generare num. interi nell'intervallo $[n, m]$ $m > n$

$m - n$ numeri

```
int l = rand()%(m-n) + n;   l ∈ [n, m-1]
```

$\in [0, \dots, m-n-1]$

moneta, dado a 2 facce: rand()%2 $\in [0, 1]$

```
l = rand()%(m-n+1) + n   l ∈ [n, m]
```

generazione razionali:

```
double x = (double) rand()/RAND_MAX;
```

$x \in [0, 1]$ $\leq$ RAND_MAX

```
x = (double) rand()/(RAND_MAX + 1.);
```

$x \in [0, 1)$

$x \in (0, 1]$

1)
```
do {
    x = (double)lrand48()/RAND_MAX;
} while (x == 0.);
```

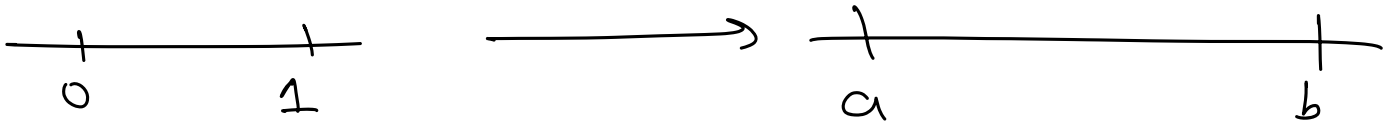
2)
```
x = 1. - lrand48()/(RAND_MAX+1.);
```

$x \in (0, 1]$

drand48(): genera $x \in [0, 1)$

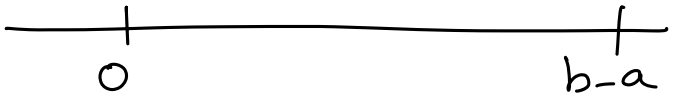
```
x = drand48();
```

problema generale: $x \in [a, b]$




```
double y = (b-a) * lrand48()/RAND_MAX
```

$y \in$

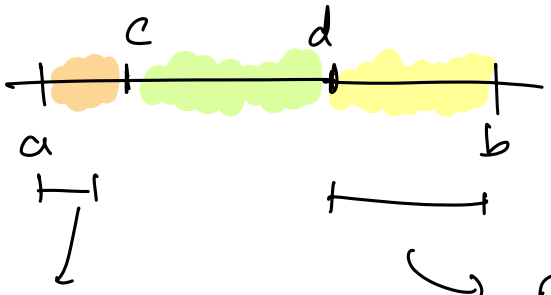


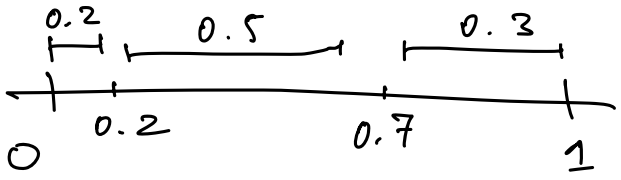

```
double z = a + (b-a) * lrand48()/RAND_MAX;
```

$z \in [a, b]$

generazione uniforme di: interi $\in [n, m]$

razionali $\in [a, b]$





meno prob.

↳ più prob.

double a = 0., b = 1.;

$x \in [a,b]$ x = a + (b-a) * (rand48()/RAND_MAX

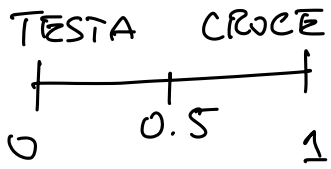
```
if( x > d ) {
    // piove
} else if( x > c ) {
    // sole
} else {
    // neve
}
```

Moneta

int moneta = lrand48() % 2; → 0 testa, 1 croce

```
#define TESTA 0
#define CROCE 1

moneta = TESTA;
if( (double) lrand48()/RAND_MAX > 0.5 ) {
    moneta = CROCE;
}
```

```
int    contessio[2] = {0};
int    monete[10000];
```




```
for (int i=0; i<10000; i++) {
    moneta = TESTA;
    if ( (double) lrand48()/RAND_MAX > 0.5 ) {
        moneta = CROCE;
    }
    monete[i] = moneta;
    contegsio[moneta] ++i
} // ciclo tiri moneta
```

Vedere # lanci

```
int nexp = 10;          1e9 ≡ 1'000'000'000
for (nexp = 10; nexp <=1e9; nexp *= 10) {
    contegsio[TESTA] = contessio[CROCE] = 0;

    for (int i=0; i<nexp ; i++) {
        moneta = TESTA;
        if ( (double) lrand48()/RAND_MAX > 0.5 ) {
            moneta = CROCE;
        }
        monete[i] = moneta;
        contegsio[moneta] ++i
    } // ciclo tiri moneta
    printf("%d esperimenti\n");
    printf("TESTA: %d CROCE: %d\n", contessio[TESTA],
                                    contessio[CROCE]);
    printf("%lf", (double) contessio[TESTA]/nexp);
} // ciclo esperimenti
```

```
R = L/2.;

do {
    x = L* lrand48()/RAND_MAX;
    y = L* lrand48()/RAND_MAX;
} while( x*x + y*y > R*R );
```