

Esercitazione individuale Verletta

Giovedì 8/1

Venerdì 9/1

} partecipazione personale
con 2 assenze max.

Settimane 8-12 dicembre

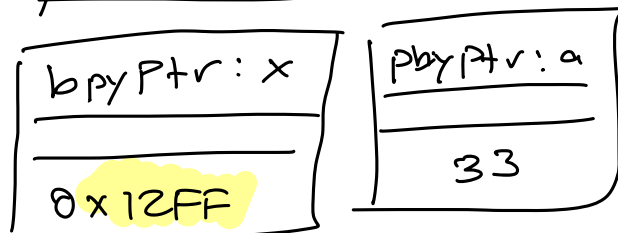
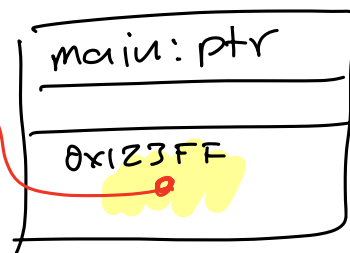
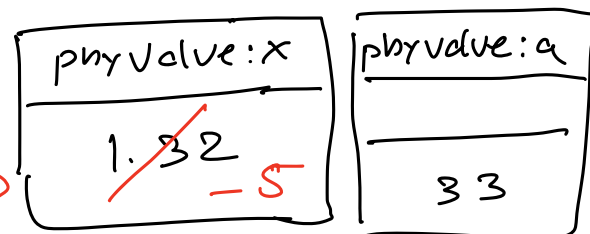
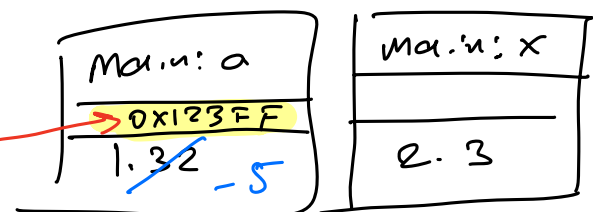
lun 8 festa

non c'è esercitazione: 8, 11, 12 dicembre

Passaggio per valore e per puntatori

```
1#include <stdlib.h>
2#include <stdio.h>
3
4void pbyValue(double);
5void pbyPtr(double*);
6
7int main() {
8
9    double a = 1.32, x = 2.3;
10   printf("main fine: \t a = %f , x = %f\n\n", a, x);
11
12   pbyValue(a);
13
14   printf("main dopo pbyValue: a = %f , x = %f\n\n", a, x);
15
16
17   double* ptr = &a;
18
19   pbyPtr(ptr);
20
21   printf("main fine: \t a = %f , x = %f\n\n", a, x);
22}
23
24void pbyValue(double x) {
25   double a = 33;
26   printf("inizio pbyValue: a = %f , x = %f\n", a, x);
27
28   x = -5.;
29   printf("fine pbyValue: a = %f , x = %f\n\n", a, x);
30}
31
32
33void pbyPtr(double* x) {
34   double a = 33;
35   printf("inizio pbyPtr: a = %f , *x = %f\n", a, *x);
36
37   *x = -5.;
38   printf("fine pbyPtr: a = %f , *x = %f\n\n", a, *x);
39}
40}
```

inizio



*x = valore delle locazione di memoria
in x.

*x è 1.32

inizio pbyPtr:

$a = 33$

$*x = 1.32$

$*x = 5$: scrivi 5 nella loc. di mem
il cui indirizzo è in x

valore di (0x123FF) diventa -5

fine pbyPtr:

$a = 33$

$*x = -5$

main fine:

$a = -5$

$x = 2.3$

```
[shamacmini:material rahatlou$ gcc -o /tmp/app passbyPtr.c
```

```
[shamacmini:material rahatlou$ /tmp/app
```

```
main fine: inizio  $a = 1.320000$ ,  $x = 2.300000$ 
```

```
inizio pbyValue:  $a = 33.000000$ ,  $x = 1.320000$ 
```

```
fine pbyValue:  $a = 33.000000$ ,  $x = -5.000000$ 
```

```
main dopo pbyValue:  $a = 1.320000$ ,  $x = 2.300000$ 
```

```
inizio pbyPtr:  $a = 33.000000$ ,  $*x = 1.320000$ 
```

```
fine pbyPtr:  $a = 33.000000$ ,  $*x = -5.000000$ 
```

```
main fine:  $a = -5.000000$ ,  $x = 2.300000$ 
```

variabili
locali in pbyValue

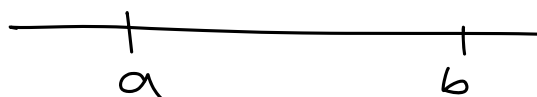
invariate
dopo pbyValue

passaggio
per
puntatore

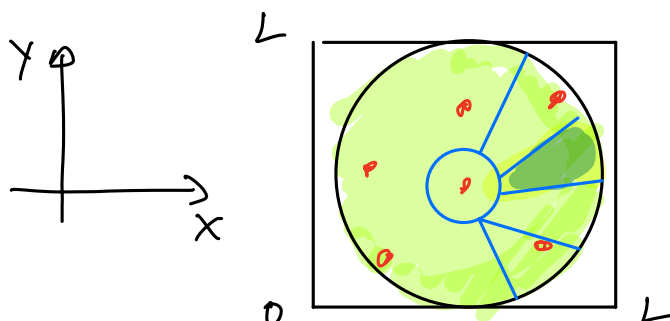
Esempio di passaggio per puntatore

double uniforme (double, double)

double $x = \text{uniforme}(-2.2, 3.3);$



genero uniforme
tra $a = -2.2$
 $b = 3.$



$R = 42$

```

nCerchio = 0;
x = uniforme(0, L)
y = uniforme(0, L)

```

// dentro o fuori?

```

if( (x - L/2.)^2 + (y - L/2.)^2 < L^2/4 ) {
    nCerchio++;
}

```

```

double probCerchio = uniforme(0, 1);
if( probCerchio >= 0.5 ) {
    nCerchio++;
}

```

generaCerchio($\frac{L}{2}$,
&x,&y);

```
do {
```

```
    x = uniforme(0, L);
```

```
    y = uniforme(0, L);
```

```

} while( (x - L/2) * (x - L/2) +
          (y - L/2) * (y - L/2)
          >= L * L / 4 );

```

```
r = sqrt( (x - L/2) * (x - L/2) + (y - L/2) * (y - L/2) );
```

```
theta = atan2(x, y)
```

In C
non si può

```
x, y = generaCerchio(R);
```

```
generaCerchio(R, &x, &y);
```

Funzione generaCerchio:

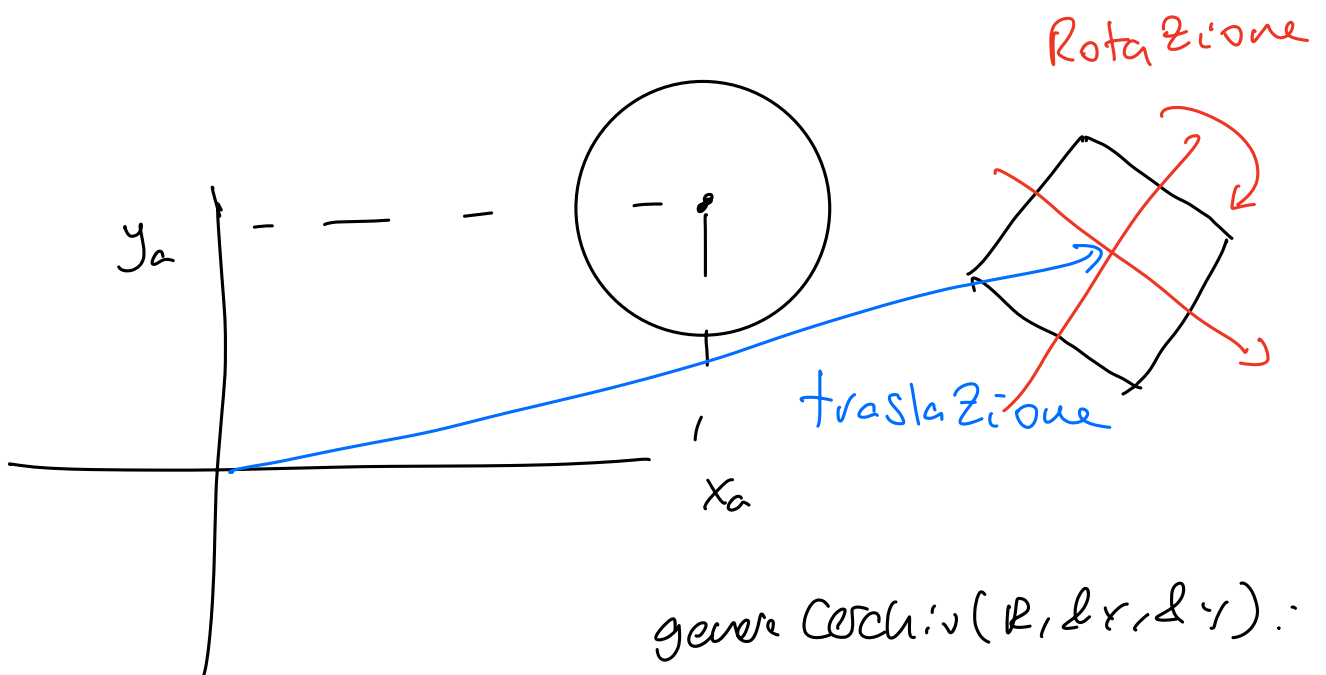
Dichiarazione: void generaCerchio(double, double*, double*);

Implementazione:

```
void generoCerchio(double R, double * x, double * y) {  
    double a, b;  
    do {  
        a = uniforme(-R, R);  
        b = uniforme(-R, R);  
    } while (a*a + b*b >= R*R);  
    *x = a;  
    *y = b;  
}
```

```
do {  
    *x = uniforme(-R, R);  
    *y = uniforme(-R, R);  
} while ((*x)*(*x) + (*y)*(*y) >= R*R);
```

$(x) \cdot (x)$



```
generoCerchio(R, &x, &y);  
x += x0;  
y += y0;
```