



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria e
Scienza dell'Informazione
DISI - Trento

Programmazione 1

07 - Esercitazione

Stefano Berlato - stefano.berlato-1@unitn.it

Anno Accademico 2023/2024

Nelle puntate precedenti

Scrivere un programma che generi un numero compreso tra 1 e 10 utilizzando la funzione presentata sotto. Chiedere poi all'utente ripetutamente di inserire un numero e terminare il programma quando si sarà indovinato.

```
#include <cstdlib>
srand(time(NULL));
int random_number = rand() % 10 + 1
```

Per maggiori informazioni:

<http://www.cplusplus.com/reference/cstdlib/rand/>

Nelle puntate precedenti

- Istruzione iterativa `for`

```
for (init; exp; agg) {  
    istruzioni  
}
```

```
for (int i=0; i<100; i++)  
{  
    cout << i << endl;  
}
```

Nelle puntate precedenti

- **Istruzione di salto** `break`, `return`, `continue`
 - `break`: esci dal loop
 - `continue`: salta l'iterazione corrente del loop
 - `return`: esci dalla funzione corrente (con valore)

00 - Let's begin!

- Tipo derivato `referimento-a (&)`
 - Consente di dare **nomi multipli** ad una variabile (in realtà, **ogni espressione dotata di indirizzo**)
 - Se la variabile originale viene modificata, allora anche il suo “alias” verrà modificato;
 - Ovviamente, se anche l’alias viene modificato allora anche la variabile originale subirà dei cambiamenti;

00 - Let's begin!

- Tipo derivato riferimento-a (&)

```
int x = 5;  
int & y = x;
```

```
x++;      // 6  
cout << y << endl;    // 6  
y++;      // 7  
cout << x << endl;    // 7
```

```
int k = 20;  
y = k;  
cout << x << endl;    // 20
```

00 - Let's begin!

- Tipo derivato `puntatore-a (*)`
 - Un puntatore contiene l'**indirizzo** di un altro oggetto precedentemente creato;
 - Ad un puntatore viene riservato uno spazio di memoria sempre uguale (atto a contenere l'indirizzo), però **non viene automaticamente riservato lo spazio per l'oggetto puntato!**
 - Per ottenere il valore dell'oggetto puntato devo utilizzare l'**operatore di dereference ***

00 - Let's begin!

- Tipo derivato puntatore-a (*)

```
int x = 5;
int * y = &x; // L'operatore & da l'indirizzo di memoria

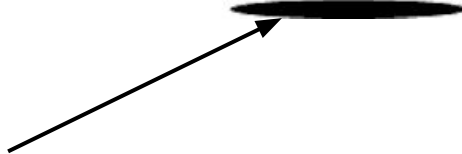
x++; // 6
cout << y << " " << *y << endl; // 6
(*y)++; // 7
cout << x << endl; // 7

int k = 20;
y = &k;
cout << x << endl; // 7
```

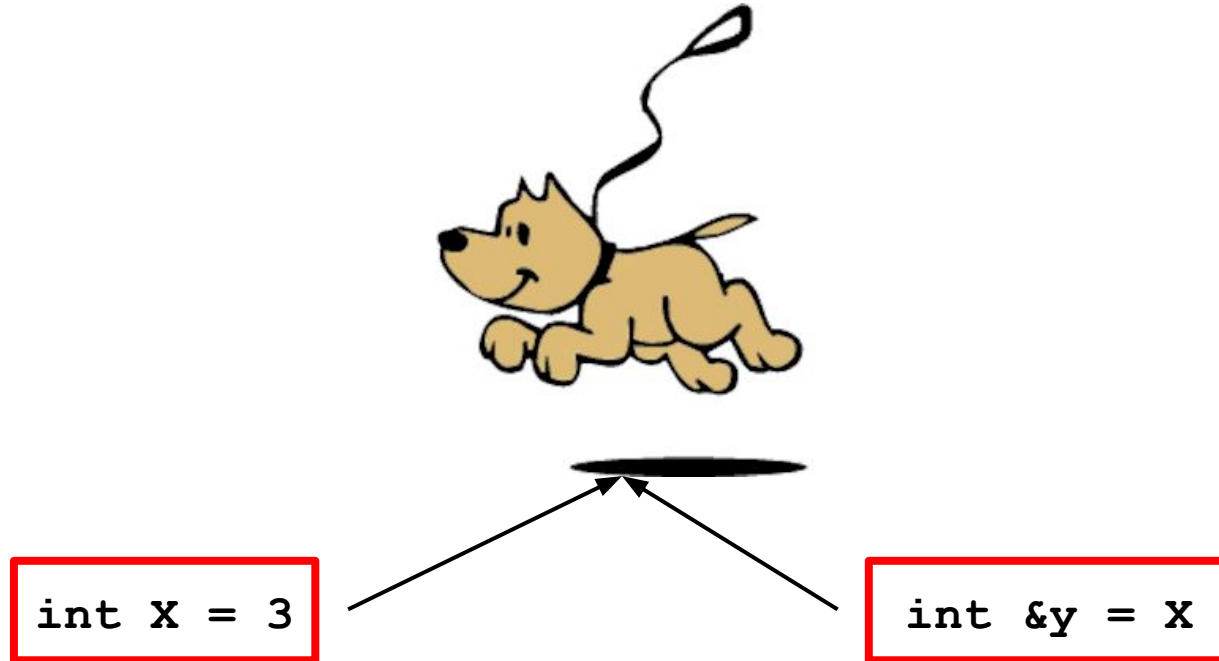

00 - Let's begin!



```
int x = 3
```



00 - Let's begin!



00 - Let's begin!

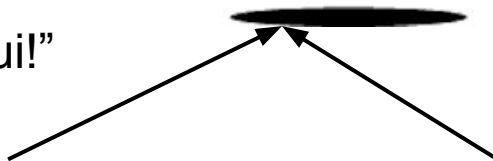


“Achille! Vieni qui!”

```
int x = 3
```

“Nerone! Vieni qui!”

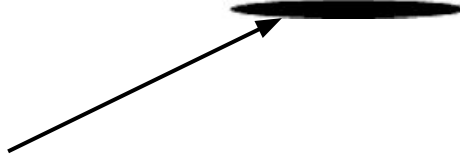
```
int &y = x
```



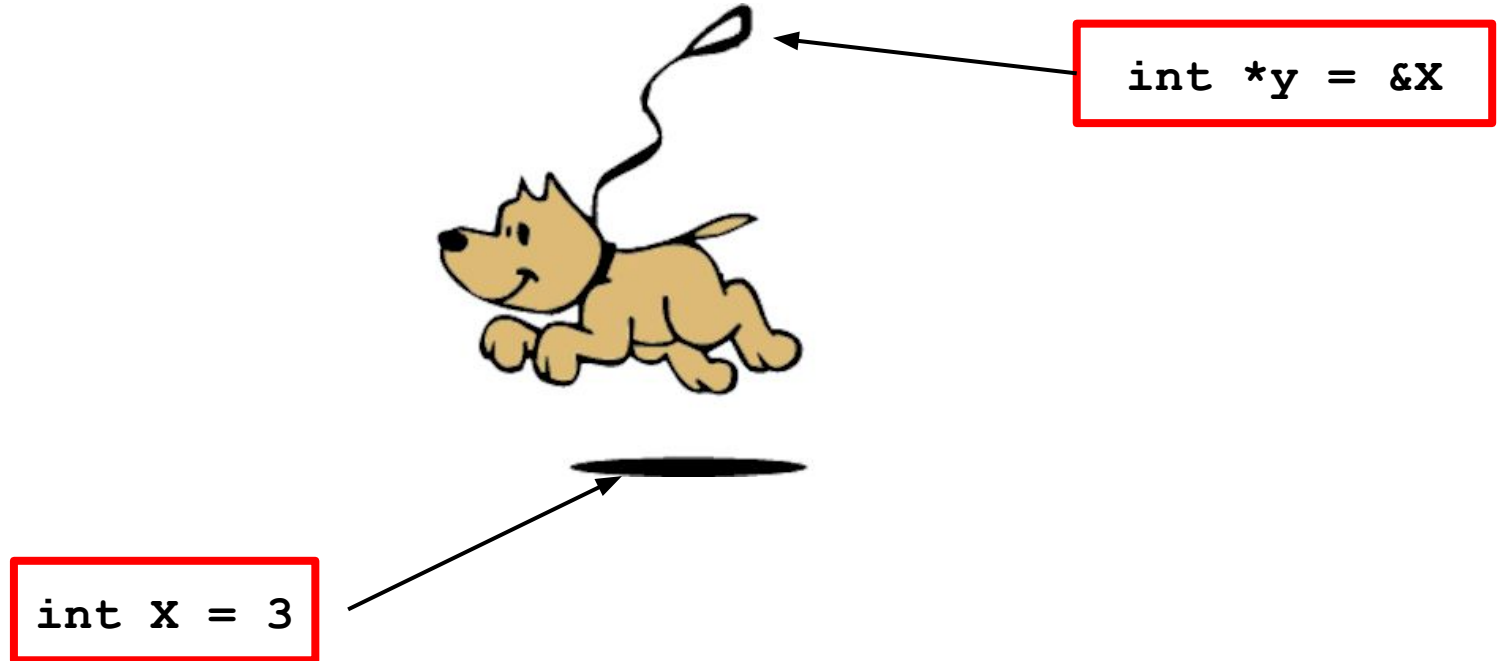
00 - Let's begin!



```
int x = 3
```



00 - Let's begin!



1 - Indovina l'output!

Quale è l'output dei seguenti tre programmi?

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int a = 32;
7.      int * ptr = &a;
8.
9.      int b = 14;
10.
11.     a += b;
12.     *ptr += b;
13.
14.     cout << a << ", " << b << endl;
15.
16.     return 0;
17. }
```

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int i = 20;
7.      int * ptr = &i;
8.      (*ptr)++;
9.
10.     int j = 15;
11.     ptr = &j;
12.     *ptr++;
13.
14.     cout << i << ", " << j << endl;
15.
16.     return 0;
17. }
```



```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int x = 5;
7.      int * y = &x;
8.      int ** k = &y;
9.
10.     // cout << x << " " << &x << endl;
11.     // cout << (*y) << " " << &y <<
endl;
12.     // cout << (*k) << " " << &k <<
endl;
13.     cout << *(*k) << endl;
14.
15.     return 0;
16. }
```

2 - ASCII: La vendetta

Scrivere un programma che stampi a video il valore ASCII di tutte le lettere dell'alfabeto minuscole e maiuscole.

3 - ASCII: Il ritorno

Scrivere un programma che stampi a video i caratteri compresi tra un intervallo di valori ASCII (quindi, interi) a e b definito dall'utente in input.

Controllare anche che l'intervallo inserito dall'utente sia valido ($a < b$, $a > 0$, $b < 127$), altrimenti terminare il programma.

4 - Somma tra primi

**Scrivere un programma che, dato in input un numero dall'utente, ritorni a video se questo numero possa essere espresso come la somma di due numeri primi.
Stampare anche i corrispondenti numeri primi a video.**

Input: 22 => Output: 17 + 5