# TIPO INTERO

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

int main ()
{
   int i;
   int i1 = 7;
   int i2 (7);
   int i3 =0, i4, i5 = 6;

   i1 = -7;
   i2 = i1 + 3;
   i2 = i1 - 1;
   i2 = i1 * 2;
   i4 = 1/2;
   i5 = 1 % 2;
   i3 = 1/2 * 2 + 1 % 2;
   cout << i3 << endl;

   return 0;
}
```

Assegnamenti

uguali

// () : SEPARATORE
// int : definisce il TIPO
// main : PAROLA CHIAVE  i = IDENTIFICATORE,
// int i : DEFINIZIONE ; int i1 = 7 : definizione con ASSEGNAMENTO

// definizione di più variabili

OPERATORI

// i1 = -7  cambiamento di segno
// i2 = -4  somma
// i2 = -8  sottrazione
// i2 = -14  moltiplicazione
// i4 = 0  quoziente
// i5 = 1  resto
// i3 = 1  (a = (a/b)*b + a % b)

// LETTERALE numeri

```
[ 1 ]
```




~ ~ ~ ~ ~ ~ ~ ~

```
# include <iostream>
using namespace std;
int main ()
{
   short int s1 = 1;
   short s2 = 2;

   long int ln1 = 654l;
   long ln2 = 6543L;
   long ln3 = 6543 l;

   int ott = 011;

   int esad1 = 0xF;

   cout << ott << endl << esad1 << endl;

   return 0;
}
```

// short int : interi corti ; letterale intero

// long int : interi lunghi

// letterale con CODIFICA IN OTTALE (prefisso 0) ; ott = 9

// letterale in esadecimale ; esad1 = 15

```
[ 9
  15 ]
```

(1)

- OPERATORI definiti:
  - Aritmetici: + ; - ; * ; / ; % (resto divisione)
  - di Confronto: == (uguale) ; != (diverso) ; > ; >= ; < ; <=
    ↳ danno risultato booleano

master copy



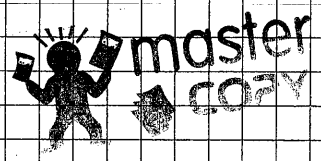

master copy

# TIPO UNSIGNED (naturali)

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int u1 = 1U;          // unsigned: codifica numeri NATURALI
    unsigned u2 = 2u;              // 1U, 2u: LETTERALI unsigned
    unsigned short int u3 = 3;
    unsigned short u4 = 4;
    unsigned long int u5 = 5555;
    unsigned long u6 = 6666UL;
    unsigned long u7 = 7777LU;
    cout << u1 << '\t' << u2 << endl;
    cout << u3 << '\t' << u4 << endl;
    cout << u5 << '\t' << u6 << '\t' << u7 << endl;

    return 0;
}
```

```
[ 1  2
  3  4
  5555  6666  7777 ]
```




- OPERATORI Bit a Bit:
  - `|` OR bit a bit
  - `&` AND bit a bit
  - `^` OR esclusivo bit a bit
  - `~` COMPLEMENTO bit a bit
  - `<<` traslazione a sinistra
  - `>>` traslazione a destra

| a | b | \| | & | ^ | $\sim_a$ | $\sim_b$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

```
#include <iostream>
using namespace std;
int main()
{
    unsigned short a = 0xFFF9;          // 1111 1111 1111 1001
    unsigned short b = ~a;              // 0000 0000 0000 0110
    unsigned short c = 0x0013;          // 0000 0000 0001 0011
    unsigned short c1, c2, c3;
    c1 = b | c;                         // 0000 0000 0001 0111
    c2 = b & c;                         // 0000 0000 0000 0010
    c3 = b ^ c;                         // 0000 0000 0001 0101
    unsigned short b1, b2;
    b1 = b << 2;                        // 0000 0000 0001 1000
    b2 = b >> 1;                        // 0000 0000 0000 0011
    cout << a << '\t' << b << '\t' << c << endl;
    cout << c1 << '\t' << c2 << '\t' << c3 << endl;
    cout << b1 << '\t' << b2 << endl;

    return 0;
}
```

| | | |
|---|---|---|
| 65529 | 6 | 19 |
| 23 | 2 | 21 |
| 24 | 3 | |

# TIPO REALE (sottoinsieme reali rappresentabili dall'elaboratore)

```
#include <iostream>
using namespace std;
int main()
{
   double d1 = 3.3;                          // double: codifica tipo REALE
   double d2 = -12.14e-3, d3 = 1.51;         // d1=3.3 : NOTAZIONE DECIMALE;
                                             // d2=... : NOTAZIONE SCIENTIFICA;
   float f = -2.2;                           // float: reali corti
   float g = f-12.12F;                       // letterale float con suffisso F
   long double h = +0.1;                     // long double: reali lunghi (n° massimo di bit)
   long double K = 1.23e+12L;                // letterale long con suffisso L
   cout << d1 << '\t' << d2 << '\t' << d3 << endl;
   cout << f << '\t' << g << endl;
   cout << h << '\t' << K << endl;

   return 0;
}
```

| | | |
|---|---|---|
| 3.3 | -0.012 14 | 1.51 |
| -2.2 | -14.32 | |
| 0.1 | 1.23e+012 | |

N.B.: se n° di cifre basta ⇒ scrittura standard decimale
se n° di cifre non basta ⇒ scrittura scientifica espon.

- OPERATORI:
  - aritmetici
  - di confronto (risultato booleano)

master copy