Espressioni booleane

Espressioni con risultato di tipo booleano

In particolare:

espressioni di confronto semplici

es.: $x > y + 1$

espressioni di confronto composte (tramite connettivi logici)

es.: $(x > 3 \ \&\& \ y < 15) \ || \ x < 0$

n.b. Alle espr. di confronto composte si applicano le normali regole dell'algebra booleana.

Ad es.: $(x == 1 \ \&\& \ y < 1.5) \ || \ (x == 1 \ \&\& \ z > 1)$

può essere riscritta equivalentemente come (applico **proprietà**

distributiva: $p \text{ and } (q \text{ or } r) \equiv (p \text{ and } q) \text{ or } (p \text{ and } r)$)

$x == 1 \ \&\& \ (y < 1.5 \ || \ z > 1)$

n.b. Espressioni valutate da sinistra a destra; non appena si determina il valore logico dell'espressione (*true* o *false*) la valutazione si ferma

es.: $-3 > 0 \ \&\& \ x < 3$ valuta solo la prima condizione e da' risultato *false*

$3 > 0 \ || \ x < 3$ valuta solo la prima condizione e da' risultato *true*

n.b. Valori booleani possono essere trattati come interi e viceversa

Booleani come interi

```
Es.:    bool b = false;  
        cout << b;           // output: 0  
        cout << b + 1;      // output: 1
```

Interi come booleani

```
Es.:    int x;  
        cin >> x;  
        if (x % 2) cout << numero dispari;  
        else cout << numero pari;
```

se x è un numero pari $x \% 2$ è uguale a 0 e quindi **false** come valore booleano