



Scuola di Ingegneria Industriale  
Laurea in Ingegneria Energetica  
Laurea in Ingegneria Meccanica



**POLITECNICO  
DI MILANO**

Dipartimento di  
Elettronica, Informazione  
e Bioingegneria



# Informatica B

## Sezione D

*Franchi Alessio Mauro, PhD    [alessiomauro.franchi@polimi.it](mailto:alessiomauro.franchi@polimi.it)*



# Elementi di logica

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

A or B è vera se almeno uno tra A e B è vero. È falso solo quando sono entrambi falsi

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

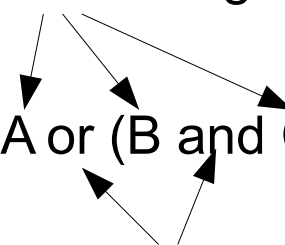
A and B è vera sse sono entrambe vere; è falsa negli altri casi.

A	Not A
0	1
1	0

Inverto il valore

Si possono costruire **espressioni logiche**. Esempio: A or (B and C)

Variabili logiche



Operatori logici

Precedenze: NOT precede AND che precede OR



# Elementi di logica

**Tautologia:** un'espressione logica sempre vera;  $A \text{ or not } A$   
**Contraddizione:** un'espressione logica sempre falsa;  $A \text{ and not } A$

Alcune proprietà dell'algebra di Boole:

I legge di De Morgan:  $\text{not } (A \text{ and } B) = \text{not } A \text{ or not } B$

II legge di De Morgan:  $\text{not } (A \text{ or } B) = \text{not } A \text{ and not } B$

Proprietà associativa, commutativa, distributiva, ...

Tramite queste proprietà si possono trasformare le espressioni:

$\text{not } A \text{ and } B \text{ or } A =$	(assorbimento)
$= \text{not } A \text{ and } B \text{ or } (A \text{ or } A \text{ and } B) =$	(togli le parentesi)
$= \text{not } A \text{ and } B \text{ or } A \text{ or } A \text{ and } B =$	(commutativa)
$= \text{not } A \text{ and } B \text{ or } A \text{ and } B \text{ or } A =$	(distributiva)
$= (\text{not } A \text{ or } A) \text{ and } B \text{ or } A =$	(legge dell'elemento 1)
$= \text{true and } B \text{ or } A =$	(vero and $B=B$ )
$= B \text{ or } A$	

è più semplice dell'espressione originale !



# Elementi di logica

Data la formula:

$$\text{not}(A \text{ or } B) \text{ and } (A \text{ or not}(C))$$

1) Si costruisca la sua tabella di verità.

2) Si consideri ora la condizione scritta in linguaggio C, in cui  $x$  e  $y$  siano due variabili di tipo int:

$$!(x > 4 \parallel x < 9) \&\& (x > 4 \parallel !(x == y))$$

ottenuta dalla prima formula sostituendo la variabile  $A$  con  $x > 4$ ,  $B$  con  $x < 9$  e  $C$  con  $x == y$ .

Si risponda alle seguenti domande:

(i) La condizione è vera o falsa quando  $x$  vale 1 e  $y$  vale 2?

(ii) La condizione è vera per qualsiasi valore di  $x$  e  $y$ ?

(iii) La condizione è falsa per qualsiasi valore di  $x$  e  $y$ ?

(c) Semplificare l'espressione nelle variabili  $A$ ,  $B$ ,  $C$  usando le proprietà dell'algebra di Boole.



# Elementi di logica

Assumendo che `c1` e `c2` siano due variabili di tipo `char`, che memorizzano rispettivamente i valori `'e'` ed `'m'`, indicare nella seguente tabella, per ognuna delle espressioni logiche

- se l'espressione è vera o falsa (per i valori delle variabili sopra indicati),
- se è sempre vera per qualsiasi valore che le due variabili possono assumere,
- se è sempre falsa per qualsiasi valore che le due variabili possono assumere.

Si fornisca una giustificazione per ogni risposta inserita nella tabella

```
((c1 != 'e') && (c2 == 'm')) || ((c1 != 'h') && (c2 == 'm')) )  
(c1 < 'g') || !( (c1 <= 'g') && (c1 != 'g') )  
(c1 <= 'm') || ( (c2 > 'm') || !(c2 > c1) )
```



# Richiamo di codifica binaria

Convertire il numero -1313.3125 nel formato IEEE 32bit (1+8+23);  
Convertire il numero 0.1015625 nel formato IEEE 32bit (1+8+23);



# Programmazione c

```
#include <stdio.h>
#define DIM 3
int main()
{
    int i,j, M[DIM][DIM];
    int s[3]={0,0,0};
    for (i=0; i<DIM; i++)
        for (j=0; j<DIM; j++)
            scanf("%d",&M[i][j]);
    for (i=0;i<DIM;i++)
    {
        s[0] += M[i][0];
        s[1] += M[i][1];
        s[2] += M[i][2];
    }
    printf("%d %d %d\n", s[0],s[1],s[2]);
    return 0;
}
```

## Es 1) Domande:

1. Cosa fa questo programma?
2. Se l'input fosse:

111222333

Quale sarebbe l'output?



# Programmazione C

```
#include <stdio.h>
#define MAX 20
int main()
{
    typedef int numeri[MAX];
    numeri A;
    int p,j,i=0;
    int s=0;
    do{
        scanf("%d",&A[i]);
        i++;
    }while(i<MAX && A[i-1]>0);
    do{
        scanf("%d",&p);
    }while(p<0||p>=i);
    for(j=0;j<p;j++)
        s=s+A[j];
}
```

## 2) Domande

1. Cosa fa questo programma?

2. Supponete che l'input sia

4 5 12 12 3 3 3 3 7 14 0 4

Quale sarà l'output?

```
for(j=p;j<i;j++)
    s=s-A[j];
if(s==0)
    printf("%d",p);
else if(s<0)
    for(j=p;j<i;j++)
        printf("%d\n",A[j]);
else
    for(j=0;j<p;j++)
        printf("%d\n",A[j]);
```





# Programmazione C

```
#include <stdio.h>
```

```
#define N 10
```

```
main()
```

```
{
```

```
    int r, c, mat[N][N];
```

```
    mat[0][0] = 1;
```

```
    for ( c=1 ; c<N ; c++ )
```

```
        mat[0][c] = 0;
```

```
    for ( r=1 ; r<N ; r++ )
```

```
    {
```

```
        mat[r][0] = 1;
```

```
        for ( c=1 ; c<N ; c++ )
```

```
            mat[r][c] = mat[r-1][c-1] + mat[r-1][c];
```

```
    }
```

```
    for ( r=0 ; r<N ; r++ )
```

```
    {
```

```
        for ( c=0 ; c<=r ; c++ )
```

```
            printf("%6d", mat[r][c]);
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

## Esercizio 3

1. Cosa fa questo programma?
2. Come si può migliorare l'output?



# Programmazione C

- 40) Scrivere un programma che legga un numero naturale da tastiera e lo converta in binario
- 41) Scrivere un programma che legga una array da tastiera e ne elimini i duplicati.
- 42) Sia  $M$  una matrice di dimensione  $N \times N$  ( $N$  costante simbolica pari a 3) di numeri interi appartenenti all'intervallo  $[0,9]$ . Si scriva un programma in grado di individuare il valore intero  $num$  che compare più frequentemente nella matrice  $M$  e di visualizzare a schermo il contenuto della matrice  $M$ , mostrando però un trattino ("--") al posto dei valori interi diversi da  $num$ .