

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Algoritmi Iterativi con Cicli, Logica Booleana e Mappe di Karnaugh

Alessandro A. Nacci
alessandro.nacci@polimi.it



POLITECNICO
DI MILANO



(Scorsa esercitazione) Esercizio 5: trova il maggiore

Trovare il maggiore tra N numeri positivi inseriti da tastiera

Pseudocodice:

- **Richiedere** quanti numeri si vogliono inserire
- **Acquisire** il numero di numeri da inserire (= contatore)
- **Finché** ho ancora numeri da inserire
 - **Richiedo** un numero
 - **Acquisisco** il numero
 - **se** è il più grande che ho visto fino ad ora
 - è il mio nuovo **massimo**

(Scorsa esercitazione) Esercizio 5: trova il maggiore

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

```
int main () {
```

```
    return 0;
```

```
}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    return 0;

}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (    ){

    }

    printf("Il massimo e' = %d \n", max);

    return 0;

}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;

    int cont=0;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (cont < N){

    }

    printf("Il massimo e' = %d \n", max);

    return 0;

}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;

    int cont=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (cont < N){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d",&val);

    }

    printf("Il massimo e' = %d \n", max);

    return 0;

}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;
    int max=0;
    int cont=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (cont < N){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d",&val);
        if (val > max)

    }

    printf("Il massimo e' = %d \n", max);

    return 0;

}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;
    int max=0;
    int cont=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (cont < N){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d",&val);
        if (val > max)
            max = val;
    }

    printf("Il massimo e' = %d \n", max);

    return 0;
}
```

(Scorsa esercitazione) Esercizio 5: trova il maggiore

```
#include <stdio.h>

int main () {

    int N;
    int max=0;
    int cont=0;
    int val;

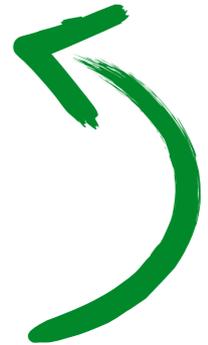
    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    while (cont < N)
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d",&val);
        if (val > max)
            max = val;
        cont++;

    printf("Il massimo e' = %d \n", max);

    return 0;

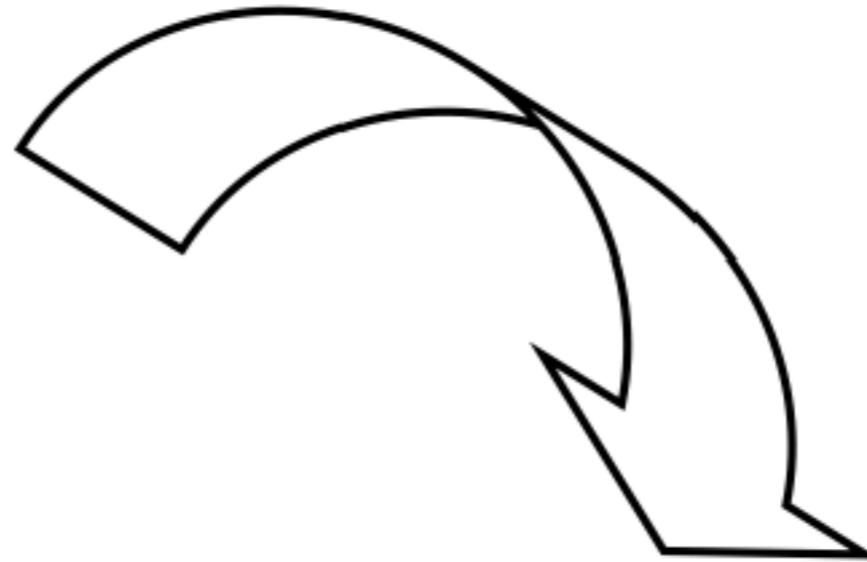
}
```



Si può fare in un altro modo?

Il ciclo for

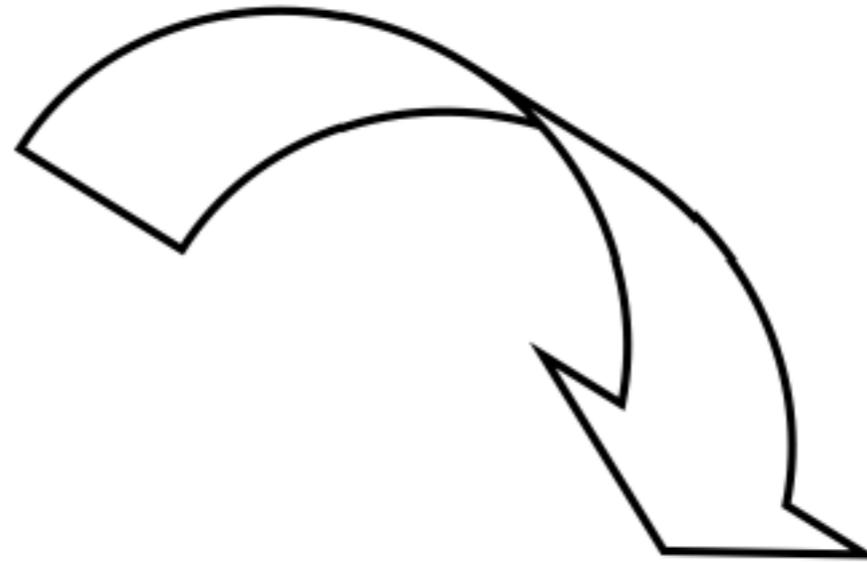
```
cont = 0;  
while cont < N {  
    ...;  
    ...;  
    cont++;  
}
```



```
for (cont = 0; cont < N; cont++) {  
    ...;  
    ...;  
}
```

Il ciclo for

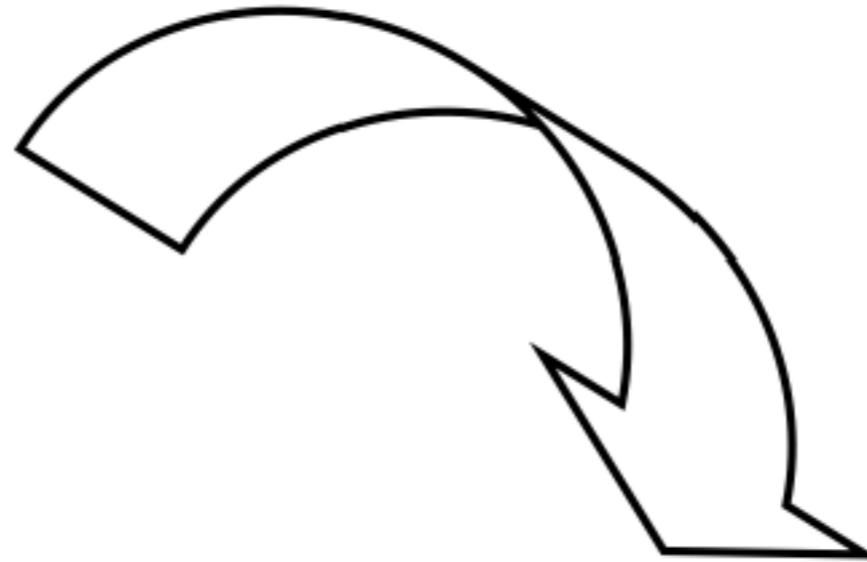
```
cont = 0;  
while (cont < N) {  
    ...;  
    ...;  
    cont++;  
}
```



```
for (cont = 0; cont < N; cont++) {  
    ...;  
    ...;  
}
```

Il ciclo for

```
cont = 0;  
while (cont < N) {  
    ...;  
    ...;  
    cont++;  
}
```



```
for (cont = 0; cont < N; cont++) {  
    ...;  
    ...;  
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>
int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);
```

```
    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>

int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);

    for(                ){

    }

    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>

int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);

    for(          cont < N;          ){

    }

    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>

int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);

    for(          cont < N;          ){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d", &val);
        if (val > max)
            max = val;
    }
    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>

int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);

    for(int cont = 0; cont < N; cont++){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d", &val);
        if (val > max)
            max = val;
    }
    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Esercizio 1: trova il maggiore con ciclo for

```
#include <stdio.h>

int main () {
    int N;
    int max=0;
    int val;

    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &N);

    for(int cont = 0; cont < N; cont++){
        printf("Inserisci il nuovo numero (positivo): ");
        scanf("%d", &val);
        if (val > max)
            max = val;
    }
    printf("Il massimo e' = %d \n", max);
    return 0;
}
```

Cicli a confronto

```
while (cont < N){  
    printf("Inserisci il nuovo numero (positivo): ");  
    scanf("%d",&val);  
    if (val > max)  
        max = val;  
    cont++;  
}
```

```
for(int cont = 0; cont < N; cont++){  
    printf("Inserisci il nuovo numero (positivo): ");  
    scanf("%d",&val);  
    if (val > max)  
        max = val;  
}
```

Esercizio 2: fattoriale con ciclo for

- Dato n , intero positivo, si definisce n fattoriale e si indica con $n!$ il prodotto dei primi n numeri interi positivi minori o uguali di quel numero. In formule

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

- Nota:
 - $0! = 1$
 - $1! = 1$
 - $2! = 2, 3! = 6, \dots$

Esercizio 2: fattoriale con ciclo for

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Esercizio 2: fattoriale con ciclo for

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

```
int main () {
```

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf( "%d", &n);
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf( "%d", &n);
```

```
    for(           ){
```

```
    }
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf( "%d", &n);
```

```
    for(int i=n; i>0; i--){
```

```
    }
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf( "%d", &n);
```

```
    for(int i=n; i>0; i--){
```

```
        fattoriale = fattoriale * i;
```

```
    }
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf("\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf("%d", &n);
```

```
    for(int i=n; i--){
```

```
        fattoriale = fattoriale * i;
```

```
    }
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf( "\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf( "%d", &n);
```

```
    for(int i=n; i > 1; i--){
```

```
        fattoriale = fattoriale * i;
```

```
    }
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo for

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

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int main () {
```

```
    int n, fattoriale = 1;
```

```
    printf("\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf("%d", &n);
```

```
    for(int i=n; i > 1; i--){
```

```
        fattoriale = fattoriale * i;
```

```
    }
```

```
    printf("\nIl fattoriale di %d e' %d \n", n, fattoriale);
```

```
    return 0;
```

```
}
```

Esercizio 2: fattoriale con ciclo while?

$$n! := \prod_{k=1}^n k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

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

```
int main () {
```

```
    int i, n, fattoriale = 1;
```

```
    printf("\n\nDi quale numero vuoi calcolare il fattoriale? ");
```

```
    scanf("%d", &n);
```

```
    i=n;
```

```
    while (i > 1){
```

```
        fattoriale = fattoriale * i;
```

```
        i--;
```

```
    }  
    printf("\nIl fattoriale di %d e' %d \n", n, fattoriale);
```

```
    return 0;
```

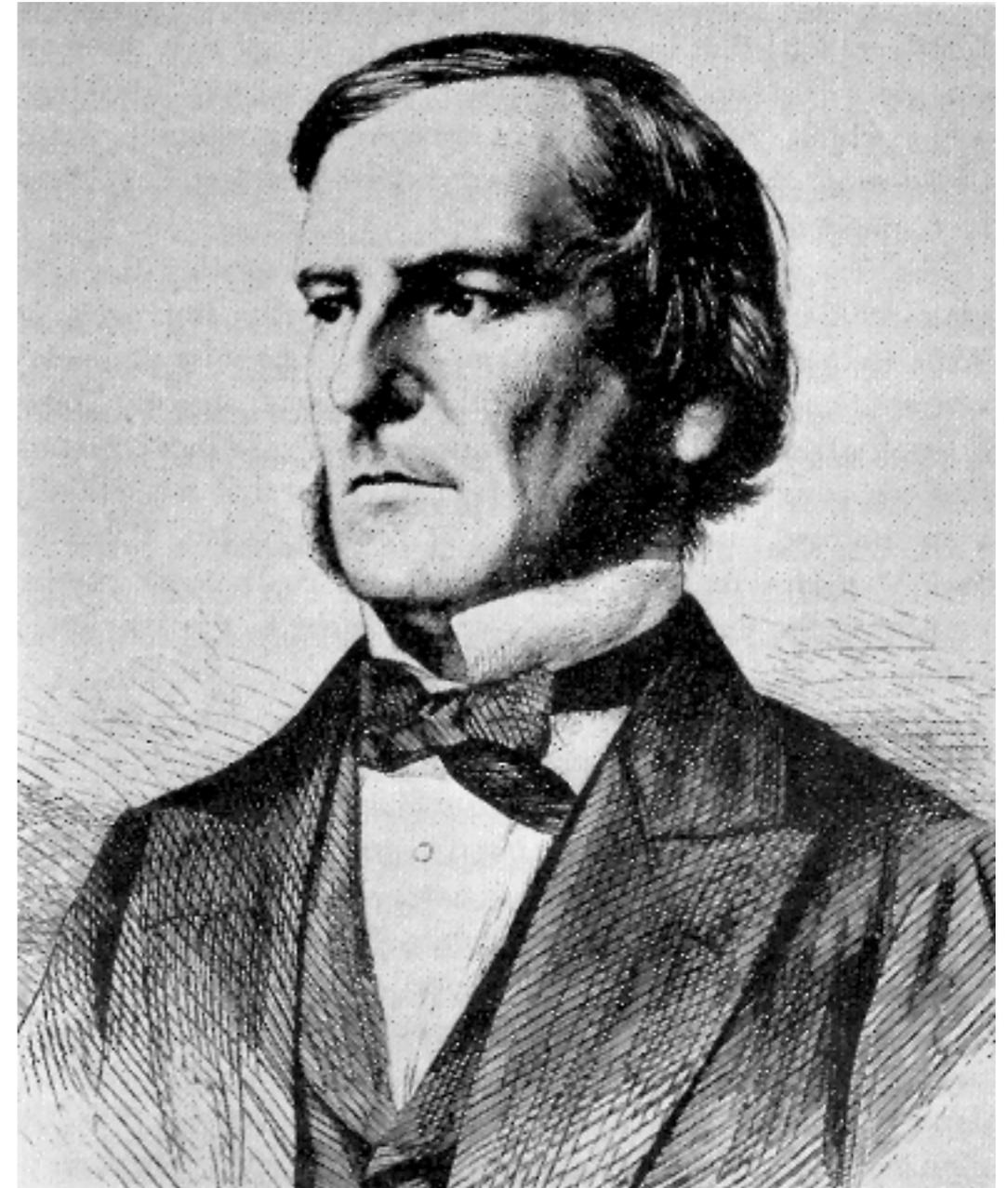
```
}
```

```
for(int i=n; i > 1; i--){  
    fattoriale = fattoriale * i;  
}
```



Logica Booleana e Mappe di Karnaugh

George Boole (Lincoln, 2 novembre 1815[1] – Ballintemple, 8 dicembre 1864[1]) è stato un matematico e logico britannico, ed è considerato il fondatore della logica matematica[1]. La sua opera influenzò anche settori della filosofia e diede vita alla scuola degli algebristi della logica.





Le operazioni logiche (*booleane*)

Operazione “and”

A	B	A and B
F	F	F
F	V	F
V	F	F
V	V	V

Operazione “or”

A	B	A or B
F	F	F
F	V	V
V	F	V
V	V	V

Operazione “not”

A	not A
F	V
V	F

Ci sono altri operatori?



Le operazioni logiche (booleane) - 2

A	NOT A
0	1
1	0



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1



A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Un riassunto **non esaustivo** su...

https://it.wikipedia.org/wiki/Algebra_di_Boole#NOT



Proprietà logiche

• • • •



Dimostrazione della proprietà di

$$A \cdot (A + B) = A$$

Dimostrazione

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot 1 + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

$$A + A \cdot B = A$$

Dimostrazione

$$A + (A \cdot B) = A \cdot 1 + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$



Leggi di De Morgan

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

p	q	\overline{p}	\overline{q}	$p \wedge q$	$\overline{p \wedge q}$	$\overline{p} \vee \overline{q}$
V	V	F	F	V	F	F
V	F	F	V	F	V	V
F	V	V	F	F	V	V
F	F	V	V	F	V	V

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

p	q	\overline{p}	\overline{q}	$p \vee q$	$\overline{p \vee q}$	$\overline{p} \wedge \overline{q}$
V	V	F	F	V	F	F
V	F	F	V	V	F	F
F	V	V	F	V	F	F
F	F	V	V	F	V	V



Esercizi sulla Algebra di Boole

ATTENZIONE!

Questi li facciamo alla lavagna ;)

Trovate delle scansioni sul mio
sito internet alessandronacci.it



Esercizio sulla Algebra di Boole

Si consideri la funzione booleana di 3 variabili $G(a, b, c)$ espressa dall'equazione seguente:

$$G(a, b, c) = abc + !a !b c + !a b c + a b !c$$

Si trasformi - tramite le proprietà dell'algebra di commutazione - l'equazione di G in modo da ridurre il costo della sua realizzazione, indicando le singole operazioni svolte e il nome oppure la forma della proprietà utilizzata.



Esercizio sulla Algebra di Boole (soluzione)

$$a b c + !a !b c + !a b c + a b !c$$



Esercizio sulla Algebra di Boole (soluzione)

$$a b c + !a !b c + !a b c + a b !c$$

$$a b c + !a !b c + !a b c + a b !c + a b c + !a b c$$

$$a b c + !a b c + !a b c + !a !b c + a b c + a b !c$$

$$(a + !a) b c + !a (b + !b) c + a b (c + !c)$$

$$1 b c + !a c$$

idempotenza

commutativa

distributiva

MOLTO CARINO MA...

Troppo intuitivo! Ergo, poco automatizzabile...

Esiste quindi un metodo più meccanico?

$$a b c + !a b c + !a c + a b$$

$$a b c + a b + !a b c + !a c$$

$$a b + !a c$$

distributiva

commutativa

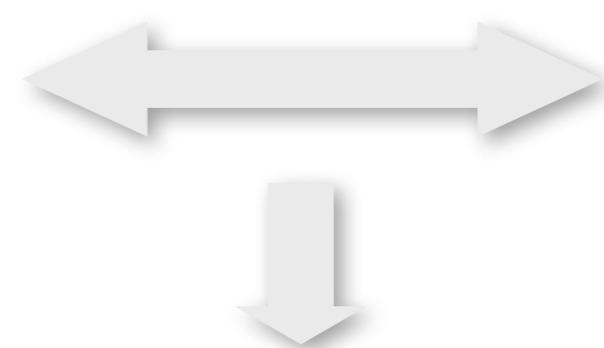
assorbimento

forma minima



Mappe di Karnaugh (I)

$$f(a,b,c) = \sum(001,011,101,110,111)$$



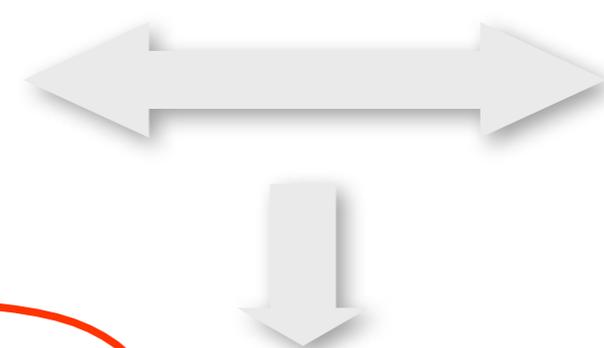
		a, b			
		00	01	11	10
c, d	00	1	1	0	0
	01	1	1	1	1
	10	0	0	1	1
	11	1	0	0	1
	10	1	0	0	1

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1



Mappe di Karnaugh (2)

$$f(a,b,c) = \sum(001,011,101,110,111)$$



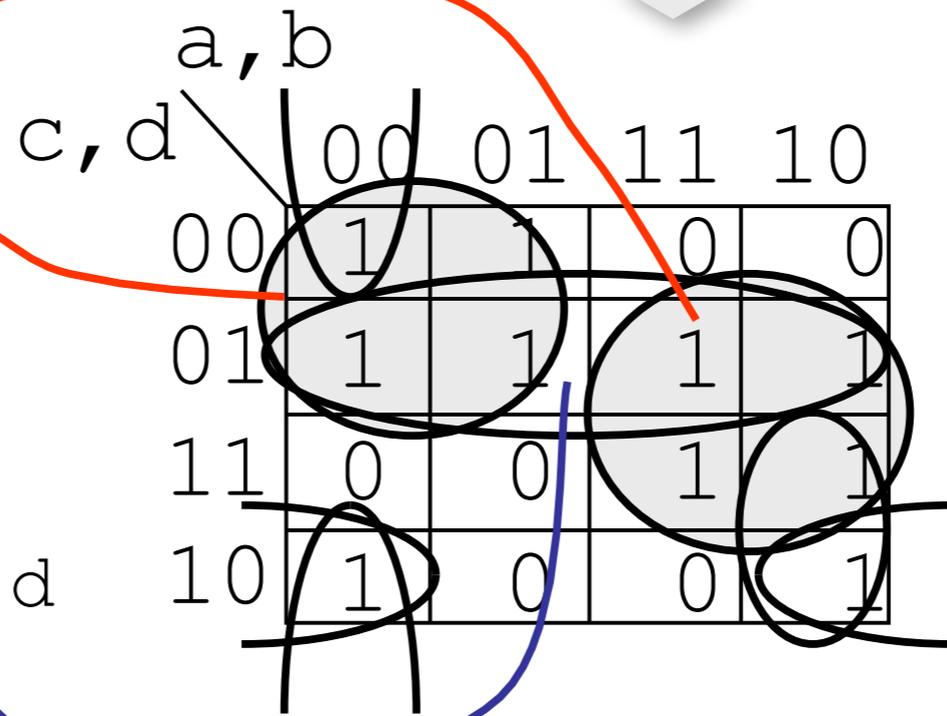
a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Implicanti primi essenziali

$a'c'$; ad

Implicanti primi

$a'b'd'$; $b'cd'$; $ab'c$; $c'd$



Completamente ridondante

Un implicante si dice implicante primo essenziale se esiste almeno un mintermine che non è coperto da nessun altro implicante primo.



Mappe di Karnaugh (3)

Tabella ottenuta dopo la selezione degli implicantanti primi essenziali

	a, b			
c, d	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

1 da coprire

Implicantanti primi essenziali

$a'c'$; ad

Implicantanti primi

$a'b'd'$; $b'cd'$; $ab'c$; ~~$c'd$~~

$f(a, b, c, d) = a'c' + ad + b'cd'$
 Forma minima (unica)

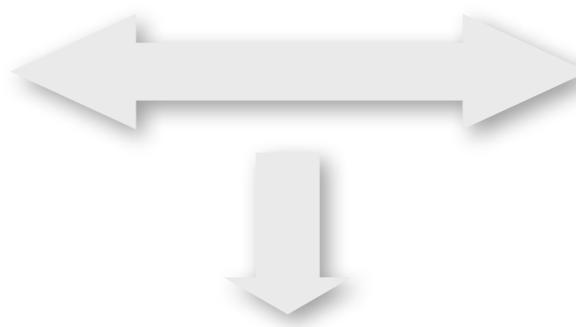
Parzialmente ridondanti

Un implicantsi si dice implicantsi primo essenziale se esiste almeno un mintermine che non è coperto da nessun altro implicantsi primo.

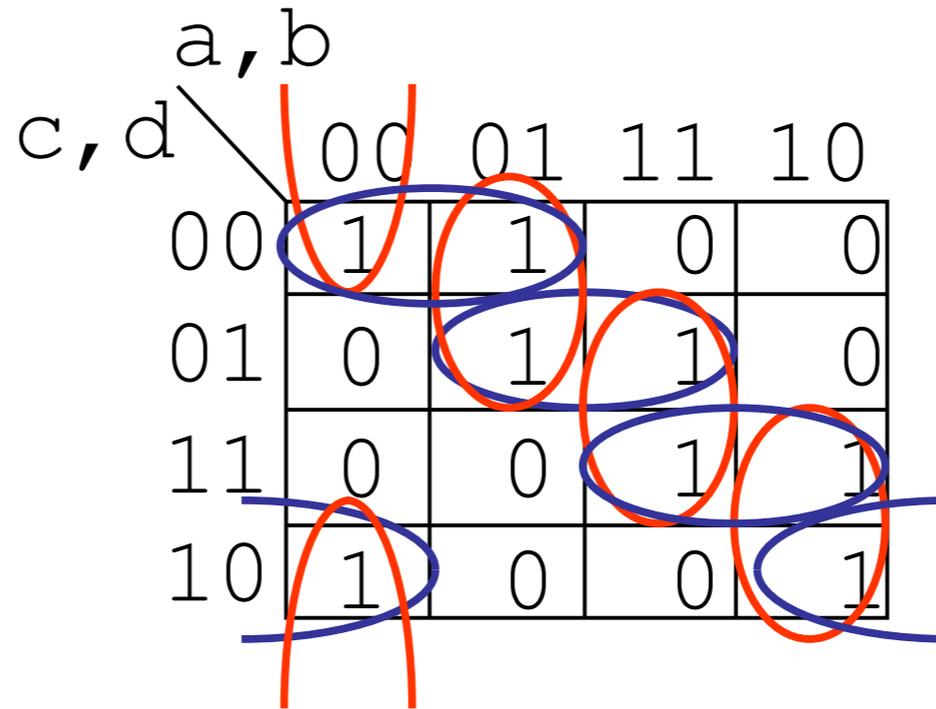


Mappe di Karnaugh (4)

$$f(a, b, c, d) = \sum (0, 2, 4, 5, 10, 11, 13, 15)$$



a	b	c	d	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1



Implicanti primi essenziali

Nessuno

Implicanti primi

$a'c'd'$; $bc'd$; acd ; $b'cd'$;
 $a'b'd'$; $a'bc'$; abd ; $ab'c$

$$f(a, b, c, d) = a'c'd' + bc'd + acd + b'cd'$$

$$f(a, b, c, d) = a'b'd' + a'bc' + abd + ab'c$$

Due forme minime

Un implicante si dice implicante primo essenziale se esiste almeno un mintermine che non è coperto da nessun altro implicante primo.



Mappe di Karnaugh (5)

- Online trovate una dispensa che le spiega con un altro esempio! :)

**Tutte il materiale sarà
disponibile sul mio sito
internet!**

www.alessandronacci.it

See You Next Time!

