

 POLITECNICO DI MILANO

Dipartimento di  
Elettronica e Informazione

# Cicli, rappresentazione binaria e complemento a due

Matteo Ferroni  
[matteo.ferroni@polimi.it](mailto:matteo.ferroni@polimi.it)

20/10/2016



POLITECNICO  
DI MILANO



# Agenda

---

- (10') Che cosa ho imparato dal 1° laboratorio
- (10') (Scorsa esercitazione) Esercizio 5: trova il maggiore
- (30') Esercizio 1: trova il maggiore con ciclo for
- (30') Esercizio 2: fattoriale con ciclo for
- (15') Pausa
- (10') Codifica binaria - ripasso
- (10') Esercizio 3 e 4 - Da decimale a binario
- (10') Esercizio 5 e 6 - Da decimale a complemento a 2
- (10') Esercizio 7 e 8 - Operazioni in complemento a 2
- (10') Esercizio 8 - Da complemento a 2 a decimale

# Che cosa ho imparato dal 1° laboratorio

---

Ok, seriamente:

1. Creare un **file vuoto** e aggiungerci l'estensione **.c**
2. Non usare degli **spazi nei nomi** dei files
3. Assegnamento ed uguaglianza: **=** e **==** sono **diversi!**
4. Input ed output: diversi **parametri** di **printf** e **scanf**
5. Il tasto '**invio**' è un carattere!
6. L'**errore** non è necessariamente dove mi dice il compilatore
7. *I cicli continuano finché non terminano...*

# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

(10') (Scorsa esercitazione) Esercizio 5: trova il maggiore

(30') Esercizio 1: trova il maggiore con ciclo for

(30') Esercizio 2: fattoriale con ciclo for

(15') Pausa

(10') Codifica binaria - ripasso

(10') Esercizio 3 e 4 - Da decimale a binario

(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# (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;

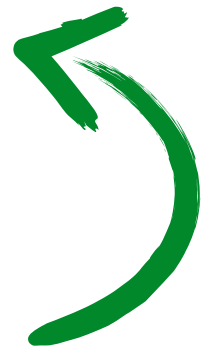
    printf("Quanti numeri vuoi inserire? ");
    scanf("%d",&N);

    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?**

# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Seconda esercitazione) Esercizio 5: trova il maggiore~~

(30') Esercizio 1: trova il maggiore con ciclo for

(30') Esercizio 2: fattoriale con ciclo for

(15') Pausa

(10') Codifica binaria - ripasso

(10') Esercizio 3 e 4 - Da decimale a binario

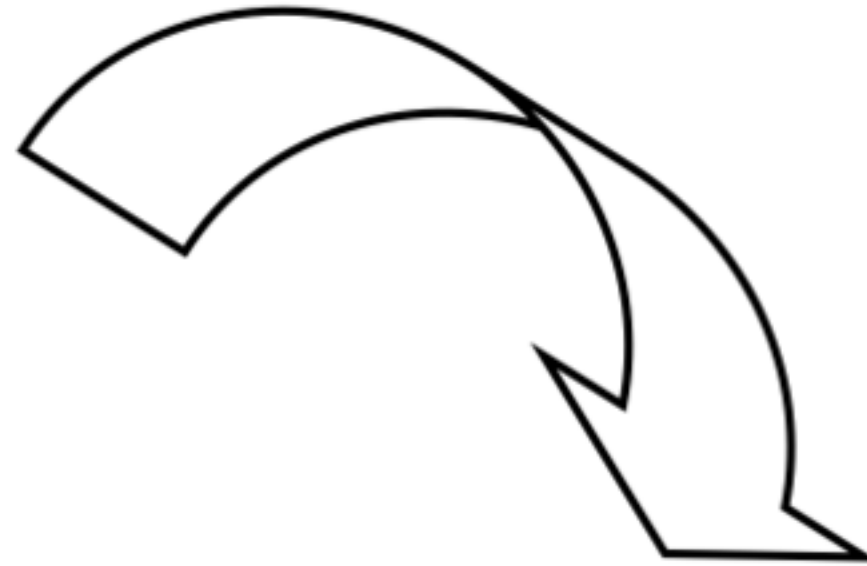
(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# Il ciclo for

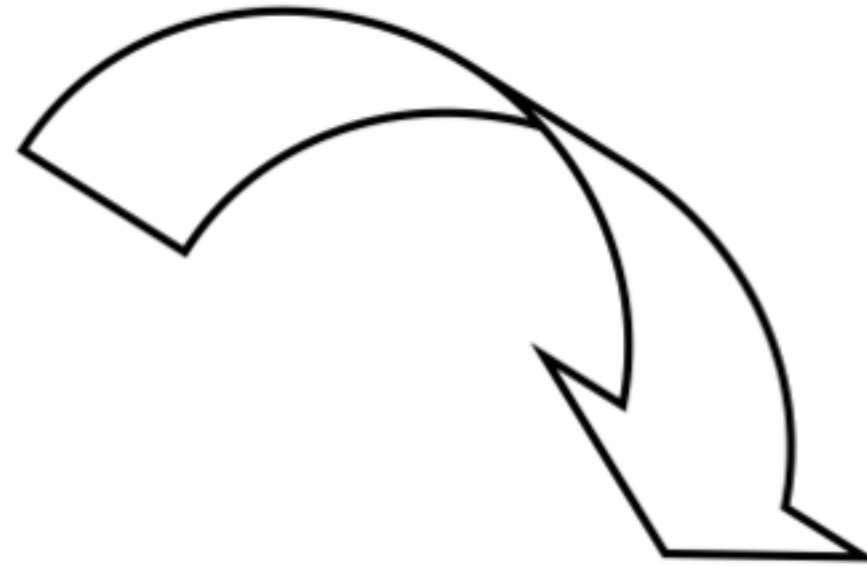
```
cont = 0;  
while (cont < N) {  
    ...;  
    ...;  
    cont++;  
}
```



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

# Il ciclo for

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

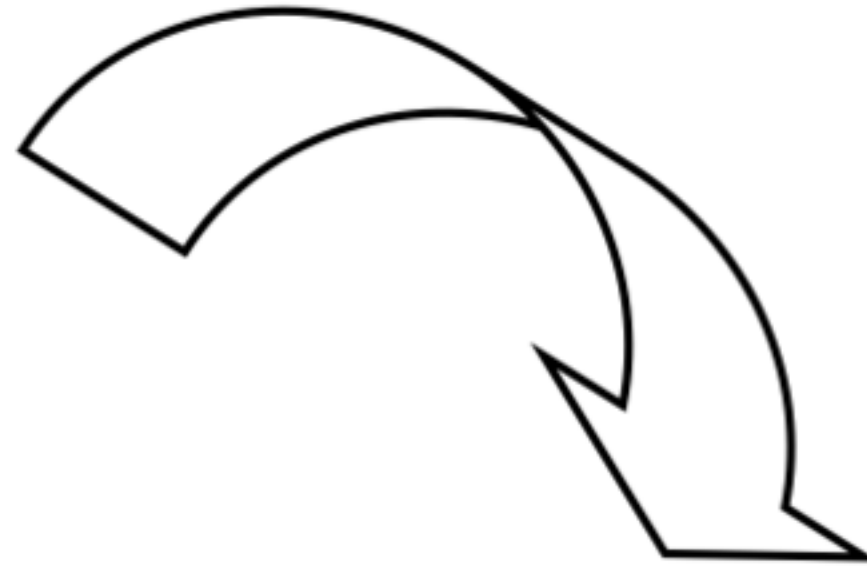


```
for (cont = 0; cont < N; cont++) {  
    ...;  
    ...;  
}
```



# 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; ){
        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;  
}
```



# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Senza esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

(30') Esercizio 2: fattoriale con ciclo for

(15') Pausa

(10') Codifica binaria - ripasso

(10') Esercizio 3 e 4 - Da decimale a binario

(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

## 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;  ){
```

```
        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;  
}
```

# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Seconda esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

~~(30') Esercizio 2: fattoriale con ciclo for~~

(15') Pausa

(10') Codifica binaria - ripasso

(10') Esercizio 3 e 4 - Da decimale a binario

(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)



# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:

# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - $1 \text{ bit} \Rightarrow 2 \text{ stati (0, 1)} \Rightarrow 2 \text{ oggetti (e.g. Vero/Falso)}$

# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti

# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti
  - 3 bit  $\Rightarrow$  8 stati (000, 001, ..., 111)  $\Rightarrow$  8 oggetti
  - ...



# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti
  - 3 bit  $\Rightarrow$  8 stati (000, 001, ..., 111)  $\Rightarrow$  8 oggetti
  - ...
  - **$k$  bit  $\Rightarrow 2^k$  stati  $\Rightarrow 2^k$  oggetti**

# Codifica binaria - Codifica

---

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti
  - 3 bit  $\Rightarrow$  8 stati (000, 001, ..., 111)  $\Rightarrow$  8 oggetti
  - ...
  - **$k$  bit  $\Rightarrow 2^k$  stati  $\Rightarrow 2^k$  oggetti**
- **Quanti bit** mi servono per codificare  $N$  oggetti:

# Codifica binaria - Codifica

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti
  - 3 bit  $\Rightarrow$  8 stati (000, 001, ..., 111)  $\Rightarrow$  8 oggetti
  - ...
  - **$k$  bit  $\Rightarrow 2^k$  stati  $\Rightarrow 2^k$  oggetti**
- **Quanti bit** mi servono per codificare  $N$  oggetti:
  - $N \leq 2^k \Rightarrow k \geq \log_2 N \Rightarrow \mathbf{k = \lceil \log_2 N \rceil}$  (intero superiore)



# Codifica binaria - Codifica

- **Alfabeto binario:** usiamo dispositivi con solo due stati
- **Problema:** assegnare un codice univoco a tutti gli oggetti compresi in un insieme predefinito (e.g. studenti)
- **Quanti oggetti** posso codificare con  $k$  bit:
  - 1 bit  $\Rightarrow$  2 stati (0, 1)  $\Rightarrow$  2 oggetti (e.g. Vero/Falso)
  - 2 bit  $\Rightarrow$  4 stati (00, 01, 10, 11)  $\Rightarrow$  4 oggetti
  - 3 bit  $\Rightarrow$  8 stati (000, 001, ..., 111)  $\Rightarrow$  8 oggetti
  - ...
  - **$k$  bit  $\Rightarrow 2^k$  stati  $\Rightarrow 2^k$  oggetti**
- **Quanti bit** mi servono per codificare  $N$  oggetti:
  - $N \leq 2^k \Rightarrow k \geq \log_2 N \Rightarrow \mathbf{k = \lceil \log_2 N \rceil}$  (intero superiore)
- **Attenzione:**  
ipotesi implicita che i codici abbiano tutti la stessa lunghezza



# Codifica binaria - Associazione

---

- Quanti sono gli oggetti compresi nell'insieme?

# Codifica binaria - Associazione

---

- Quanti sono gli oggetti compresi nell'insieme?
  - 26 lettere maiuscole + 26 minuscole  $\Rightarrow$  52
  - 10 cifre
  - Circa 30 segni d'interpunzione
  - Circa 30 caratteri di controllo (EOF, CR, LF, ...)circa 120 oggetti complessivi  $\Rightarrow$

# Codifica binaria - Associazione

---

- Quanti sono gli oggetti compresi nell'insieme?
  - 26 lettere maiuscole + 26 minuscole  $\Rightarrow$  52
  - 10 cifre
  - Circa 30 segni d'interpunzione
  - Circa 30 caratteri di controllo (EOF, CR, LF, ...)circa 120 oggetti complessivi  $\Rightarrow$   
 $k = \lceil \log_2 120 \rceil = 7$
- Codice ASCII: utilizza 7 bit e quindi può rappresentare al massimo  $2^7=128$  caratteri
  - Con 8 bit (= byte) rappresento 256 caratteri (ASCII esteso)
  - Si stanno diffondendo codici più estesi (e.g. UNICODE) per rappresentare anche i caratteri delle lingue orientali

# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Scorsa esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

~~(30') Esercizio 2: fattoriale con ciclo for~~

~~(15') Pausa~~

~~(10') Codifica binaria - ripasso~~

(10') Esercizio 3 e 4 - Da decimale a binario

(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# Codifica binaria - Da decimale a binario

---

- **Esempio: il numero  $6_{10}$ :**

# Codifica binaria - Da decimale a binario

---

- Esempio: il numero  $6_{10}$ :

$$6/2 = 3 \text{ resto } 0$$

$$3/2 = 1 \text{ resto } 1$$

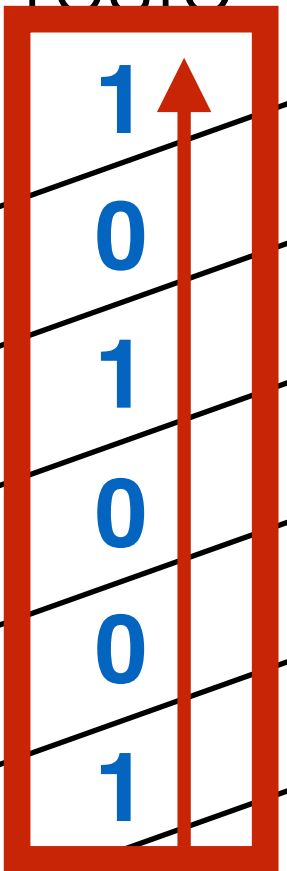
$$1/2 = 0 \text{ resto } 1$$

- Leggendo i resti dal basso verso l'alto, si ha che la rappresentazione binaria del numero  $6_{10}$  è  $110_2$
- Per una corretta verifica basta riconvertire il risultato alla base 10
  - Cioè, calcolare  $1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

# Esercizio 3 - Da decimale a binario

Come si scrive 37 in binario?

divido per due	resto	risultato
37	1	18
18	0	9
9	1	4
4	0	2
2	0	1
1	1	0
0		



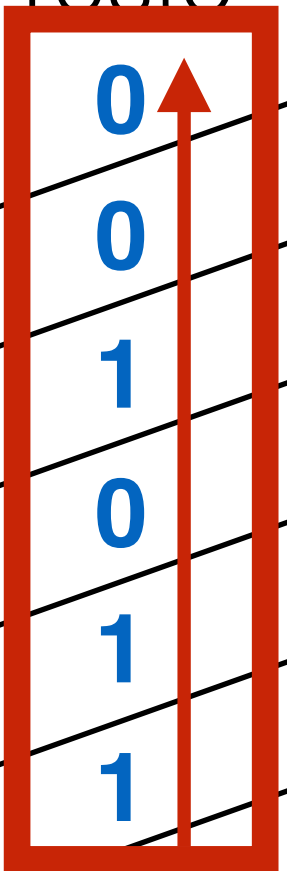
37 in binario è **100101**



# Esercizio 4 - Da decimale a binario

Come si scrive 52 in binario?

divido per due	resto	risultato
52	0	26
26	0	13
13	1	6
6	0	3
3	1	1
1	1	0
0		



52 in binario è **110100**



# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Scorsa esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

~~(30') Esercizio 2: fattoriale con ciclo for~~

~~(15') Pausa~~

~~(10') Codifica binaria - ripasso~~

~~(10') Esercizio 3 e 4 - Da decimale a binario~~

(10') Esercizio 5 e 6 - Da decimale a complemento a 2

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# Codifica binaria - Da decimale a complemento a 2

---

# Codifica binaria - Da decimale a complemento a 2

---

- Il bit più significativo rappresenta il segno del numero: 0 per i numeri positivi e 1 per i numeri negativi

# Codifica binaria - Da decimale a complemento a 2

---

- Il bit più significativo rappresenta il segno del numero: 0 per i numeri positivi e 1 per i numeri negativi
- La rappresentazione di un numero positivo si ottiene codificando il valore assoluto del numero con i bit restanti

# Codifica binaria - Da decimale a complemento a 2

---

- Il bit più significativo rappresenta il segno del numero: 0 per i numeri positivi e 1 per i numeri negativi
- La rappresentazione di un numero positivo si ottiene codificando il valore assoluto del numero con i bit restanti
- La rappresentazione di un numero negativo si ottiene in tre passi:
  - Si rappresenta in complemento a due il numeri positivo con lo stesso valore assoluto del numero negativo da codificare
  - Si invertono tutti i bit in tale rappresentazione ( $0 \rightarrow 1, 1 \rightarrow 0$ )
  - Si somma uno al risultato ottenuto al passo precedente

# Codifica binaria - Operazioni

---

- **Addizione:**

$$0 + 0 = 0 \text{ con riporto } 0$$

$$0 + 1 = 1 \text{ con riporto } 0$$

$$1 + 0 = 1 \text{ con riporto } 0$$

$$1 + 1 = 0 \text{ con riporto } 1$$

- **Esempi:**

$$\begin{array}{r} 1 + \\ \underline{1} = \end{array}$$

# Codifica binaria - Operazioni

---

- **Addizione:**

$$0 + 0 = 0 \text{ con riporto } 0$$

$$0 + 1 = 1 \text{ con riporto } 0$$

$$1 + 0 = 1 \text{ con riporto } 0$$

$$1 + 1 = 0 \text{ con riporto } 1$$

- **Esempi:**

$$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array} \quad \begin{array}{r} 101 + \\ 11 = \\ \hline \end{array}$$

# Codifica binaria - Operazioni

---

- **Addizione:**

$$0 + 0 = 0 \text{ con riporto } 0$$

$$0 + 1 = 1 \text{ con riporto } 0$$

$$1 + 0 = 1 \text{ con riporto } 0$$

$$1 + 1 = 0 \text{ con riporto } 1$$

- **Esempi:**

$$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$$

$$\begin{array}{r} 101 + \\ 11 = \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 10110101 + \\ 1000110 = \\ \hline \end{array}$$



# Codifica binaria - Operazioni

---

- **Addizione:**

$$0 + 0 = 0 \text{ con riporto } 0$$

$$0 + 1 = 1 \text{ con riporto } 0$$

$$1 + 0 = 1 \text{ con riporto } 0$$

$$1 + 1 = 0 \text{ con riporto } 1$$

- **Esempi:**

$$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$$

$$\begin{array}{r} 101 + \\ 11 = \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 10110101 + \\ 1000110 = \\ \hline 11111011 \end{array}$$

$$\begin{array}{r} 111 + \\ 11 = \\ \hline \end{array}$$

# Codifica binaria - Operazioni

---

- **Addizione:**

$$0 + 0 = 0 \text{ con riporto } 0$$

$$0 + 1 = 1 \text{ con riporto } 0$$

$$1 + 0 = 1 \text{ con riporto } 0$$

$$1 + 1 = 0 \text{ con riporto } 1$$

- **Esempi:**

$$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$$

$$\begin{array}{r} 101 + \\ 11 = \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 10110101 + \\ 1000110 = \\ \hline 11111011 \end{array}$$

$$\begin{array}{r} 111 + \\ 11 = \\ \hline 1010 \end{array}$$

# Esercizio 5 - Da binario a complemento a 2

Come si scrive -37 in complemento a 2?

1) 37 in binario è **100101**

2) **100101**  $\longrightarrow$  **011010**

$$\begin{array}{r} 3) \text{ **011010** + } \\ \quad \quad \quad \text{1 =} \\ \hline \quad \quad \text{011011} \end{array} \longrightarrow \boxed{\text{1011011}}$$

- Si rappresenta in complemento a due il numeri positivo con lo stesso valore assoluto del numero negativo da codificare
- Si invertono tutti i bit in tale rappresentazione (0 $\rightarrow$ 1, 1 $\rightarrow$ 0)
- Si somma uno al risultato ottenuto al passo precedente

# Esercizio 6 - Da binario a complemento a 2

Come si scrive -52 in complemento a 2?

1) 52 in binario è **110100**

2) **110100**  $\longrightarrow$  **001011**

$$\begin{array}{r} 3) \text{ **001011** + } \\ \quad \quad \quad \text{1 =} \\ \hline \quad \quad \text{001100} \end{array} \longrightarrow \boxed{\text{1001100}}$$

- Si rappresenta in complemento a due il numeri positivo con lo stesso valore assoluto del numero negativo da codificare
- Si invertono tutti i bit in tale rappresentazione (0 $\rightarrow$ 1, 1 $\rightarrow$ 0)
- Si somma uno al risultato ottenuto al passo precedente

# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Scorsa esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

~~(30') Esercizio 2: fattoriale con ciclo for~~

~~(15') Pausa~~

~~(10') Codifica binaria - ripasso~~

~~(10') Esercizio 3 e 4 - Da decimale a binario~~

~~(10') Esercizio 5 e 6 - Da decimale a complemento a 2~~

(10') Esercizio 7 e 8 - Operazioni in complemento a 2

(10') Esercizio 8 - Da complemento a 2 a decimale

# Esercizio 7 - Operazioni in complemento a 2

---

Quanto fa **37 + 52** ?

$$\begin{array}{r} 100101 + \\ 110100 = \\ \hline 1011001 \end{array}$$

Come si torna in decimale?

$$1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 89$$

# Esercizio 8 - Operazioni in complemento a 2

---

Quanto fa **37 - 52** ?

$$\begin{array}{r} 0100101 + \\ 1001100 = \\ \hline 1110001 \end{array}$$

Come si torna in decimale?

# Codifica binaria - Da complemento a 2 a decimale

---

- Per ottenere un numero con segno data la sua rappresentazione in complemento a due:
  - Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale



# Codifica binaria - Da complemento a 2 a decimale

---

- Per ottenere un numero con segno data la sua rappresentazione in complemento a due:
  - Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale
  - Se il primo bit è 1 il numero è negativo:

# Codifica binaria - Da complemento a 2 a decimale

---

- Per ottenere un numero con segno data la sua rappresentazione in complemento a due:
  - Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale
  - Se il primo bit è 1 il numero è negativo:
    - Si ignora il primo bit

# Codifica binaria - Da complemento a 2 a decimale

---

- Per ottenere un numero con segno data la sua rappresentazione in complemento a due:
  - Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale
  - Se il primo bit è 1 il numero è negativo:
    - Si ignora il primo bit
    - Si invertono i restanti bit

# Codifica binaria - Da complemento a 2 a decimale

---

- Per ottenere un numero con segno data la sua rappresentazione in complemento a due:
  - Se il primo bit è 0 il numero è positivo: per calcolarne il valore assoluto si esegue la conversione da binario a decimale
  - Se il primo bit è 1 il numero è negativo:
    - Si ignora il primo bit
    - Si invertono i restanti bit
    - Si converte il numero da binario a decimale

# Esercizio 8 - Operazioni in complemento a 2

---

**37 - 52** : come si torna in decimale?

**1110001**

1) **1110001**

2) **110001**  $\longrightarrow$  **001110**

3)  $0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 14$

4)  $14 + 1 = 15 \longrightarrow$  **-15**

- Se il primo bit è 1 il numero è negativo:
  - Si ignora il primo bit
  - Si invertono i restanti bit
  - Si converte il numero da binario a decimale
  - Si somma uno al numero ottenuto per ottenere il valore assoluto del numero negativo



# Agenda

---

~~(10') Che cosa ho imparato dal 1° laboratorio~~

~~(10') (Scorsa esercitazione) Esercizio 5: trova il maggiore~~

~~(30') Esercizio 1: trova il maggiore con ciclo for~~

~~(30') Esercizio 2: fattoriale con ciclo for~~

~~(15') Pausa~~

~~(10') Codifica binaria - ripasso~~

~~(10') Esercizio 3 e 4 - Da decimale a binario~~

~~(10') Esercizio 5 e 6 - Da decimale a complemento a 2~~

~~(10') Esercizio 7 e 8 - Operazioni in complemento a 2~~

~~(10') Esercizio 8 - Da complemento a 2 a decimale~~

# Bonus - Da tema d'esame Febbraio 2016

---

Un programma deve controllare se un numero **N** inserito dall'utente sia contemporaneamente:

- primo,
- dispari,
- compreso tra 3 e 100, estremi inclusi.

Quando il numero **N** inserito non soddisfa tutte le condizioni citate, l'inserimento deve essere ripetuto.

...

4. Si mostri la correttezza, o non correttezza, della condizione riportata nel ciclo while compilando una tabella di verità, ove:

A: `primo(N)`

B: `(3 <= N)`

C: `(N <= 100)`

D: `(mod(N, 2))`



# Bonus - Da tema d'esame Febbraio 2016

---

La condizione  $(3 \leq N)$  implica che il numero  $N$  sia maggiore di 3, la condizione `primo(N)` richiede che il numero sia primo, la condizione  $(\text{mod}(N,2))$  e' invece inutile; infatti un numero primo e' sicuramente dispari, a meno che non sia pari a 2, ma questo valore viene escluso dalla condizione  $(3 \leq N)$ .

L'espressione da valutare e' quindi: `!(primo(N)&&(3 <= N)&&(N <= 100))`.

	A	B	C	(A && B)	(A && B && C)	!(A && B && C)
	0	0	0			
A: primo(N)	0	0	1			
B: (3 <= N)	0	1	0			
C: (N <= 100)	0	1	1			
D: (mod(N,2))	1	0	0			
	1	0	1			
	1	1	0			
	1	1	1			

# Bonus - Da tema d'esame Febbraio 2016

La condizione  $(3 \leq N)$  implica che il numero  $N$  sia maggiore di 3, la condizione  $\text{primo}(N)$  richiede che il numero sia primo, la condizione  $(\text{mod}(N,2))$  e' invece inutile; infatti un numero primo e' sicuramente dispari, a meno che non sia pari a 2, ma questo valore viene escluso dalla condizione  $(3 \leq N)$ .

L'espressione da valutare e' quindi:  $!(\text{primo}(N) \&\& (3 \leq N) \&\& (N \leq 100))$ .

	A B C	(A && B)	(A && B && C)	!(A && B && C)
	0 0 0	0		
A: primo(N)	0 0 1	0		
B: (3 <= N)	0 1 0	0		
C: (N <= 100)	0 1 1	0		
D: (mod(N,2))	1 0 0	0		
	1 0 1	0		
	1 1 0	1		
	1 1 1	1		

# Bonus - Da tema d'esame Febbraio 2016

La condizione  $(3 \leq N)$  implica che il numero  $N$  sia maggiore di 3, la condizione  $\text{primo}(N)$  richiede che il numero sia primo, la condizione  $(\text{mod}(N,2))$  e' invece inutile; infatti un numero primo e' sicuramente dispari, a meno che non sia pari a 2, ma questo valore viene escluso dalla condizione  $(3 \leq N)$ .

L'espressione da valutare e' quindi:  $!(\text{primo}(N) \&\& (3 \leq N) \&\& (N \leq 100))$ .

	A B C	(A && B)	(A && B && C)	!(A && B && C)
	0 0 0	0	0	
A: primo(N)	0 0 1	0	0	
B: (3 <= N)	0 1 0	0	0	
C: (N <= 100)	0 1 1	0	0	
D: (mod(N,2))	1 0 0	0	0	
	1 0 1	0	0	
	1 1 0	1	0	
	1 1 1	1	1	

# Bonus - Da tema d'esame Febbraio 2016

La condizione  $(3 \leq N)$  implica che il numero  $N$  sia maggiore di 3, la condizione  $\text{primo}(N)$  richiede che il numero sia primo, la condizione  $(\text{mod}(N,2))$  e' invece inutile; infatti un numero primo e' sicuramente dispari, a meno che non sia pari a 2, ma questo valore viene escluso dalla condizione  $(3 \leq N)$ .

L'espressione da valutare e' quindi:  $!(\text{primo}(N) \&\& (3 \leq N) \&\& (N \leq 100))$ .

	A B C	(A && B)	(A && B && C)	!(A && B && C)
	0 0 0	0	0	1
A: primo(N)	0 0 1	0	0	1
B: (3 <= N)	0 1 0	0	0	1
C: (N <= 100)	0 1 1	0	0	1
D: (mod(N,2))	1 0 0	0	0	1
	1 0 1	0	0	1
	1 1 0	1	0	1
	1 1 1	1	1	0