



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Fondamenti di Informatica

Ing. Alba Amato, PhD

alba.amato@unina2.it



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

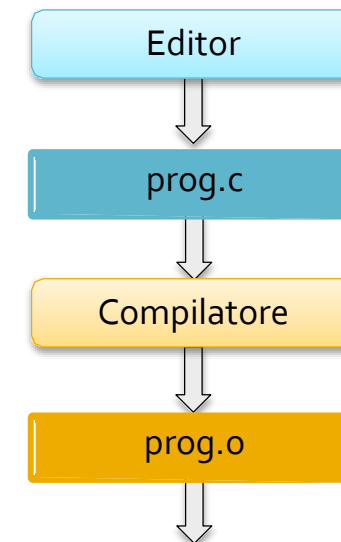
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

RIEPILOGO ED ESERCITAZIONI

Lucidi tratti dal Web



- Per essere eseguito da una CPU, un algoritmo deve essere espresso in linguaggio macchina
 - Insieme di istruzioni che la CPU comprende (sequenze di bit)
- C e C++ sono linguaggi *di alto livello*
 - La loro potenza espressiva è superiore a quella del linguaggio macchina
 - Pensati per aiutare i programmatori
- Per tutti i linguaggi di programmazione esiste
 - La **grammatica**
 - Il programma di **traduzione** in linguaggio macchina
- Traduzione
 - **Sorgente**: linguaggio da cui parte la traduzione
 - **Oggetto**: linguaggio in cui si traduce



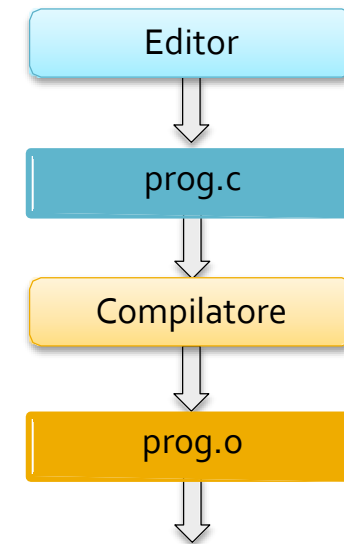


○ Compilazione

- La traduzione avviene una volta sola
 - Velocità di esecuzione
 - Il programma dipende strettamente dalla CPU per la quale è stato prodotto

○ Interpretazione

- La traduzione avviene ogni volta che il programma viene eseguito
 - Tempi più lunghi
 - Maggiore portabilità



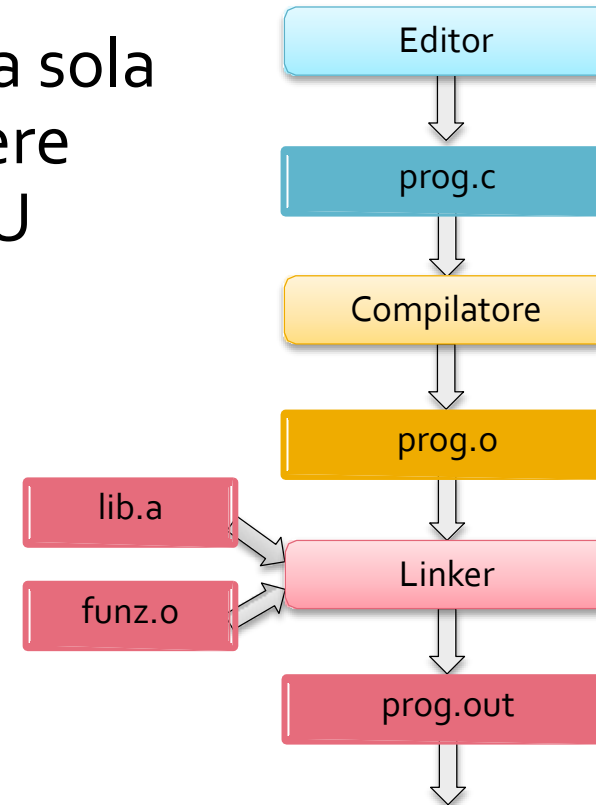


○ Nell' approccio per compilazione la sola traduzione non è sufficiente a rendere il programma «eseguibile» dalla CPU

- Servono delle funzionalità raccolte in librerie
 - Interazione con il sistema operativo
 - Gestione dell'I/O

○ Linker

- Assembla tutti gli oggetti e librerie necessari per generare un programma che sia eseguibile dalla CPU

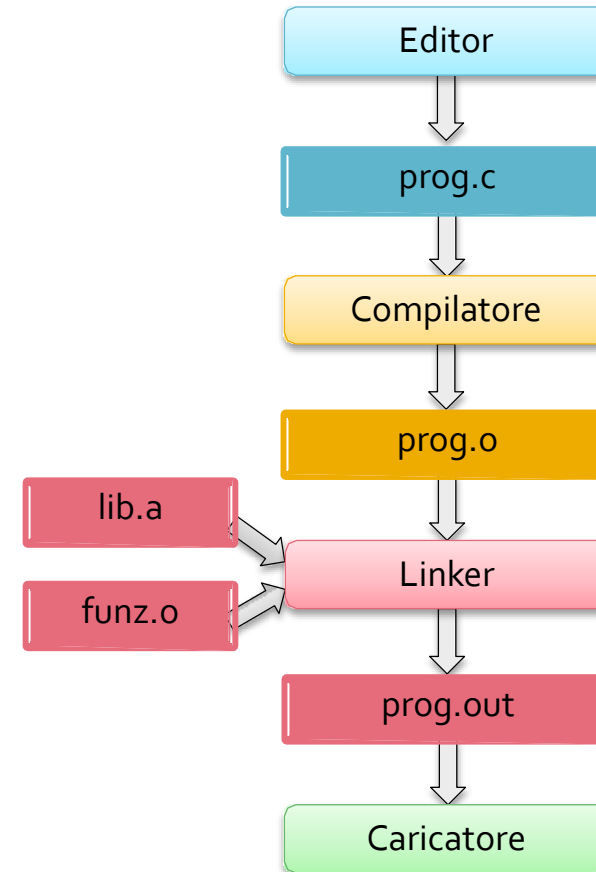




- Loader o Caricatore
 - Carica in memoria il programma
 - Attiva il programma

- Il suffisso dei file dipende dal sistema operativo ospitante
 - *Windows*
 - Sorgente: .c / .cpp
 - Oggetto: .obj
 - Libreria: .lib
 - Esecuibile: .exe

 - *Unix*
 - Sorgente: .c / .cpp
 - Oggetto: .o
 - Libreria: .a
 - Esecuibile: .out





○ Ambienti integrati di sviluppo (IDE)

- In essi sono presenti funzionalità per
 - Scrivere e modificare codice sorgente (editor)
 - Gli editor sono sensibili alle parole chiave
 - ★ Codice più facile da leggere
 - Tradurre il codice sorgente (compilatore)
 - Generare l'eseguibile (linker)
 - Effettuare il debugging
- Utilizzeremo **CodeBlocks**
 - IDE FREE che funziona da front-end verso il compilatore gcc pienamente compatibile con lo standard ANSI C
 - Sito web:
<http://www.codeblocks.org/>

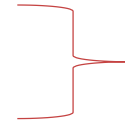


○ Gestione degli errori

■ Classificazione

■ Errori che si verificano

- Nella compilazione del sorgente
- Nel collegamento di oggetti e librerie
- Nel caricamento dell'eseguibile in memoria
- Nell'esecuzione



Il programma non viene eseguito

■ Errori più frequenti

■ Nella compilazione

- Assenza del ";" per chiudere le istruzioni
- Mancanza di parentesi () per una funzione senza parametri
- Errori dovuti al Case Sensitive

■ Nel collegamento

- Uso di un elemento in un sorgente diverso dal modo in cui è definito in un altro sorgente

■ Nel caricamento

- Un programma necessita di più memoria di quella esistente

■ Nell'esecuzione (anche chiamati eccezioni)

- Tipicamente sono errori logici
 - ★ Es.: divisione per 0, operazioni ripetute all'infinito



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

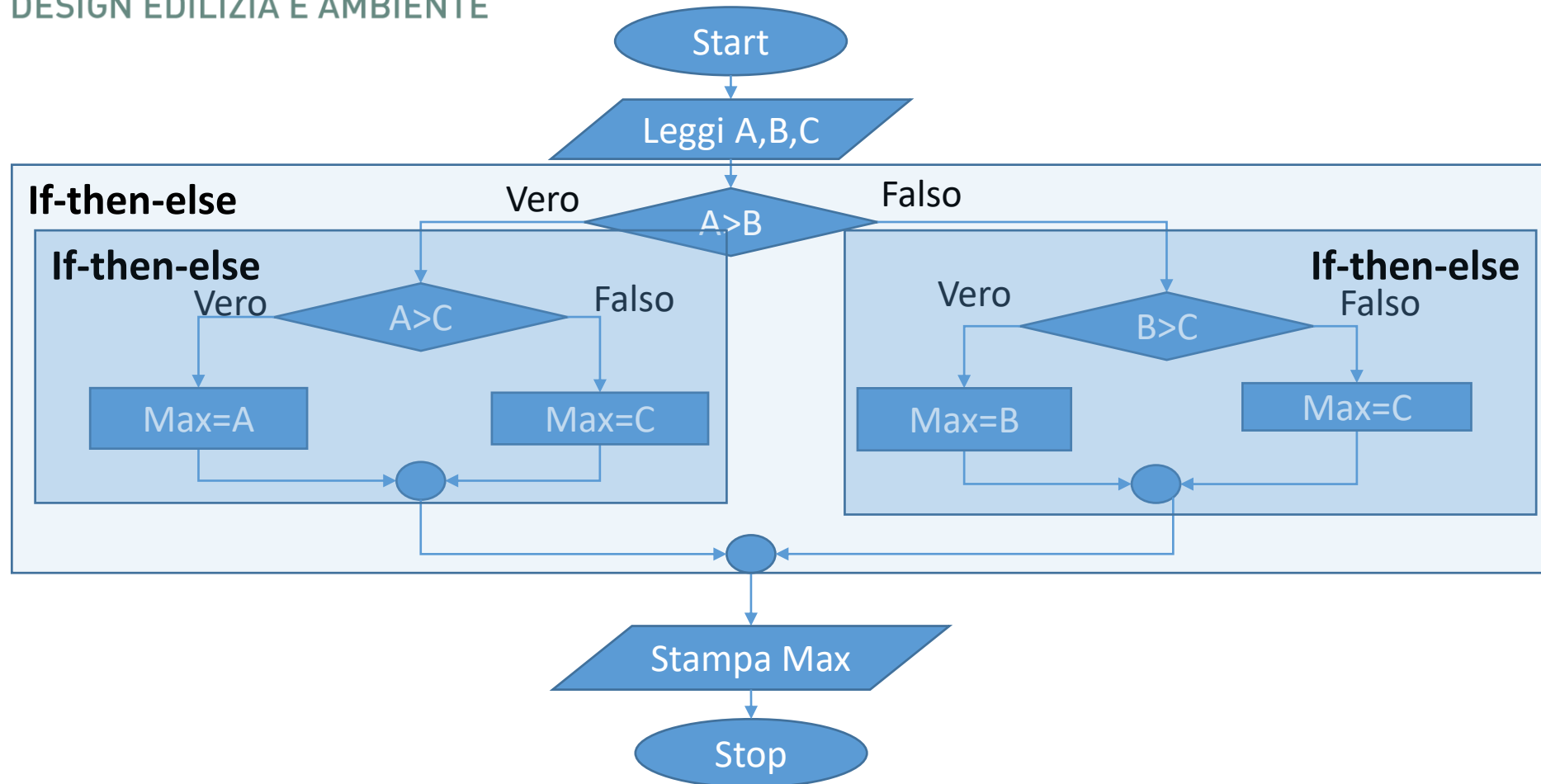
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 1

Richiedere in ingresso 3 numeri e stampare il massimo



Esempio 1





Esempio 1

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

```
int main()
```

```
{
```

```
    int a,b,c;
```

```
    printf("inserisci a: ");
```

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

```
    printf("inserisci b: ");
```

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

```
    printf("inserisci c: ");
```

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

```
if(a > b) {
```

```
    if (a > c)
```

```
        printf ("massimo= %d\n",a);
```

```
    else
```

```
        {printf ("massimo= %d\n",c);}
```

```
    }
```

```
else if(b > c)
```

```
    {printf ("massimo= %d\n",b);}
```

```
else
```

```
    {printf ("massimo= %d\n",c);}
```

```
}
```



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

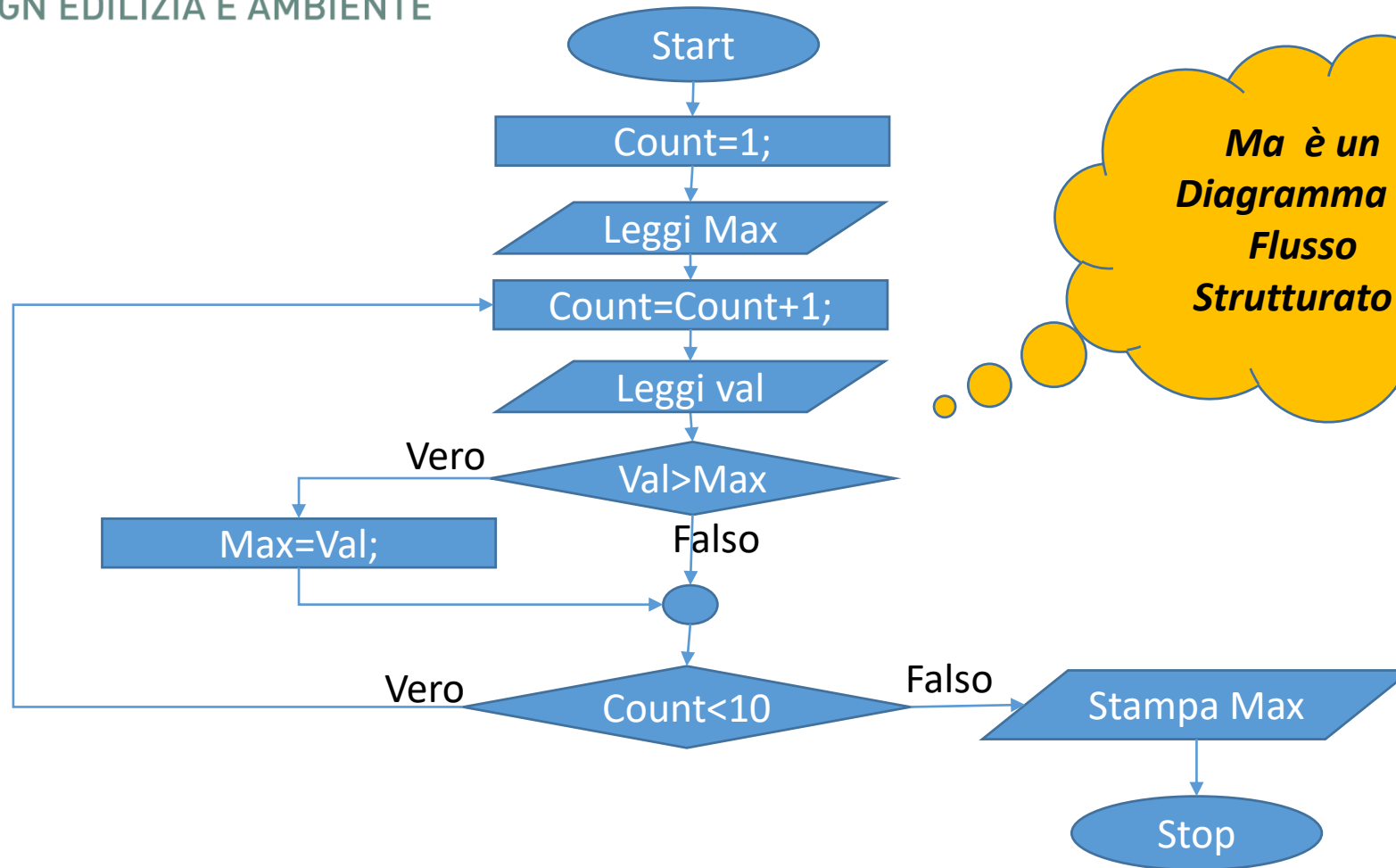
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 2

Richiedere in ingresso 10 numeri e stampare il massimo



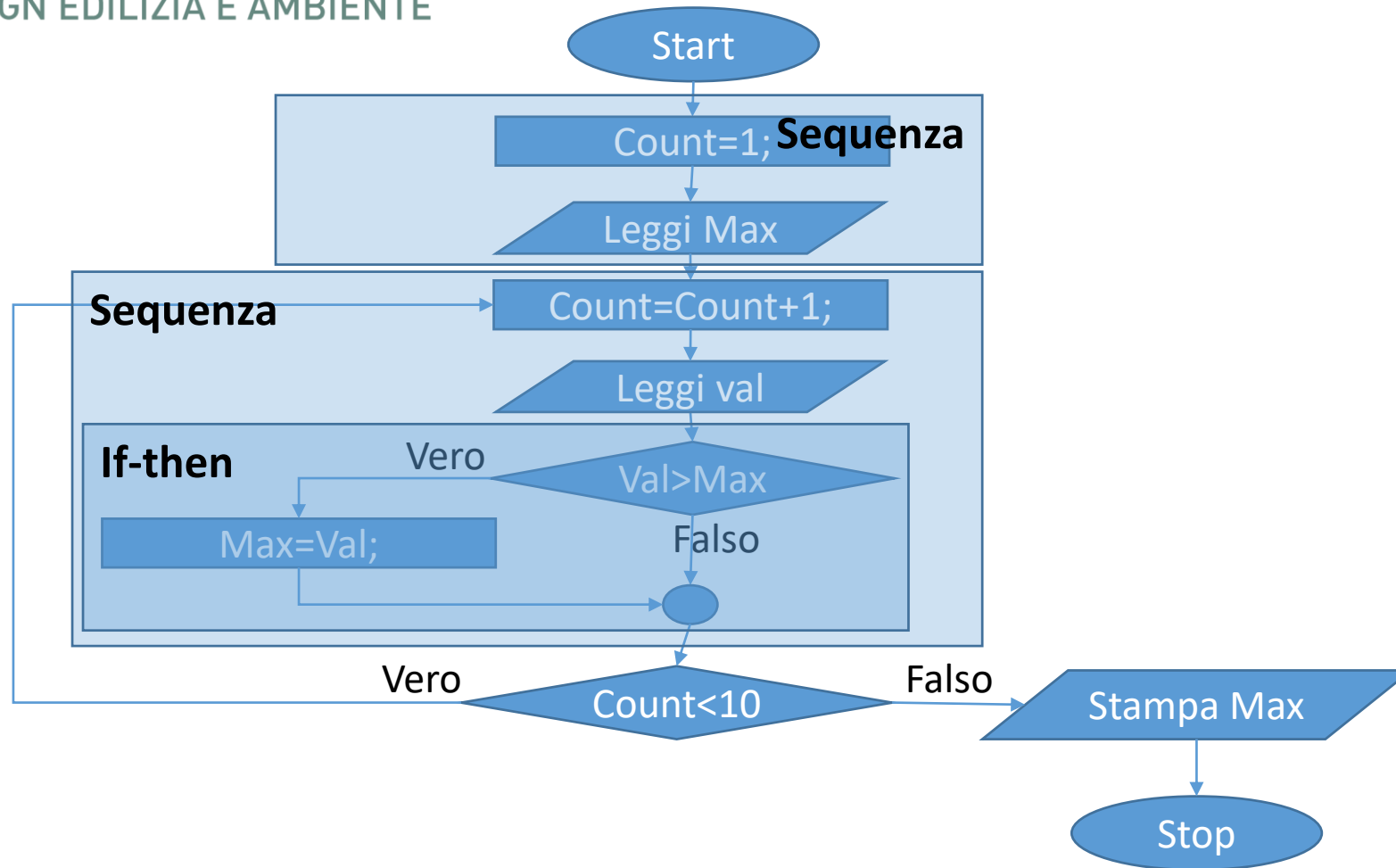
Esempio 2



*Ma è un
Diagramma di
Flusso
Strutturato?*

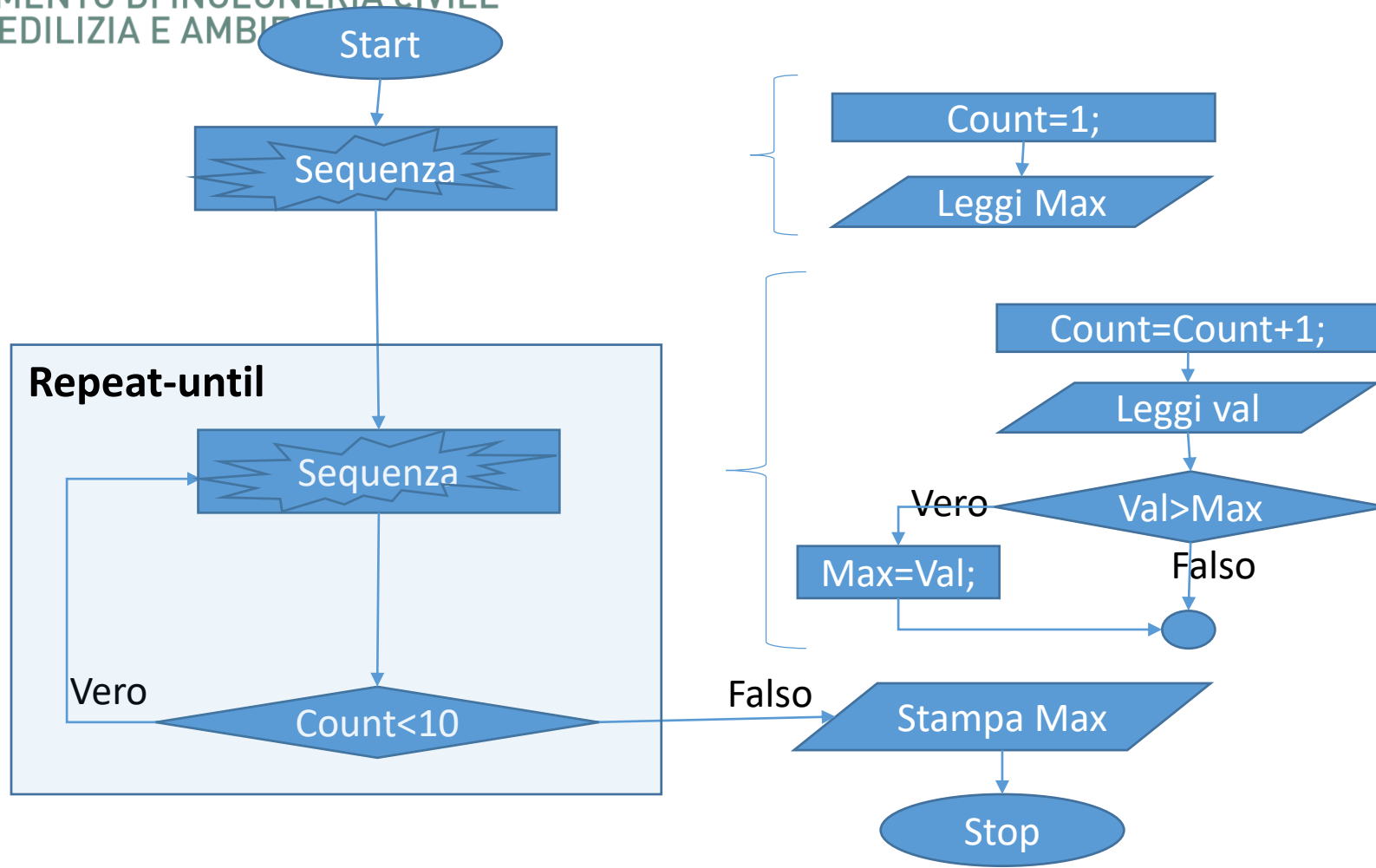


Esempio 2





Esempio 2





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 2

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

```
int main()
```

```
{
```

```
    int n;
```

```
    float a, max;
```

```
    do
```

```
    {
```

```
        printf("Quanti numeri vuoi inserire ? ");
```

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

```
        if(n<0)
```

```
            printf("Deve essere n >= 0\n");
```

```
    }
```

```
    while(n<0);
```

```
    for (int i=0;i<n;i++)
```

```
    {
```

```
        printf("inserisci il numero %d-iesimo : ",i);
```

```
        scanf("%f",&a);
```

```
        if(i==0)
```

```
            max = a;
```

```
        else if(a > max)
```

```
        {
```

```
            max=a;
```

```
        }
```

```
    }
```




UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 3

***Realizzare il Diagramma di Flusso
dell'Algoritmo per il calcolo del Massimo
Comun Divisore***



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 3

Teorema di Euclide:

“ogni divisore comune di a e b è divisore di a , b e del resto r della divisione tra a e b ($a \bmod b$), se questo non è nullo”

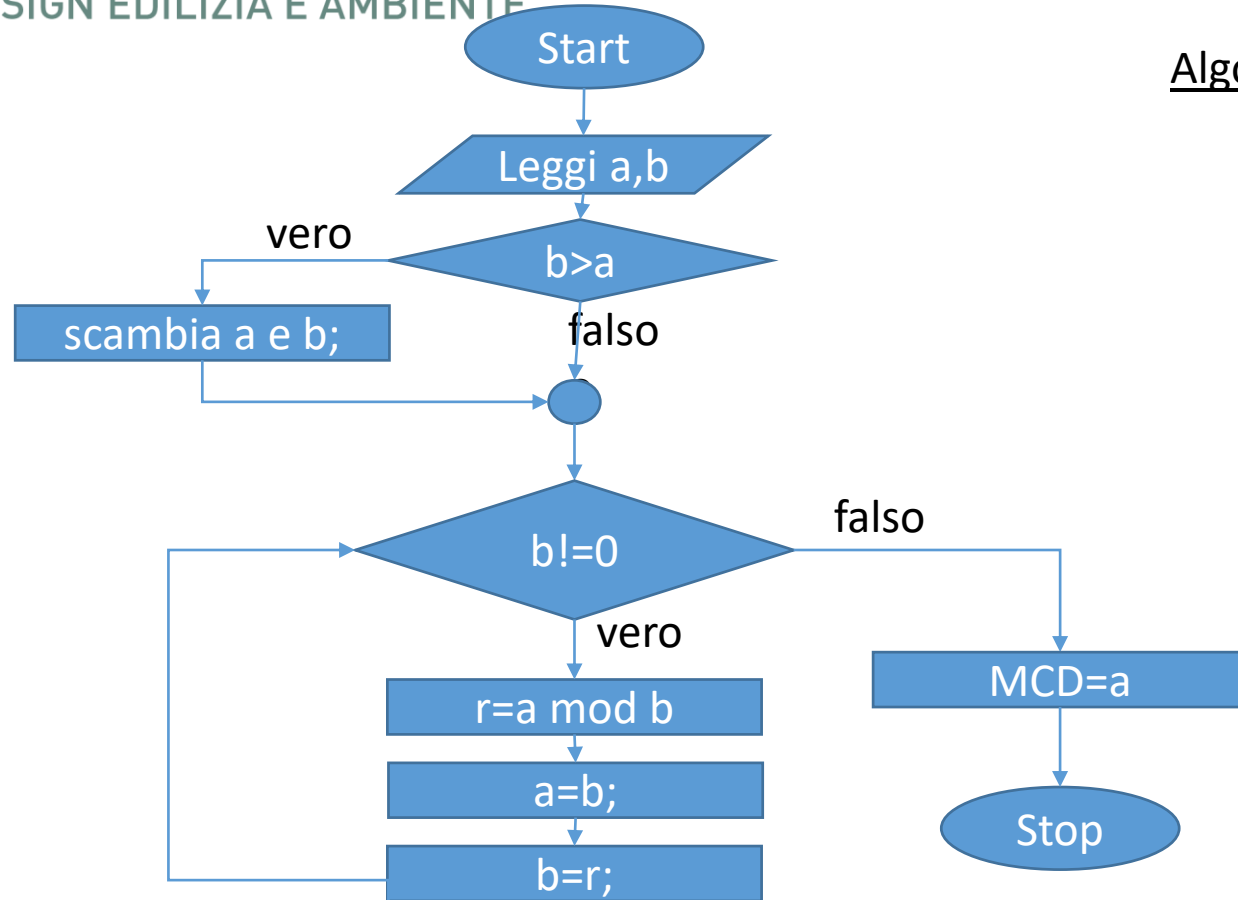
Soluzione di Euclide - Costruzione dell'algoritmo

Dati due numeri a, b assumiamo che a sia sempre il maggiore

Effettuiamo l'operazione $r = a \bmod b$ se $r = 0$ allora b è MCD di a

Altrimenti calcoliamo $r' = b \bmod r$, se $r' = 0$ allora r è divisore di b ed è MCD di a (non lo era b)

Possiamo quindi procedere così fino a trovare il caso in cui l'operazione dia zero.



Esempio 3

Algoritmo:

1. acquisire due numeri a,b
2. se $b > a$ scambiare a con b
3. se $b = 0$ $MCD(a,b) = a$ e termina
4. $r = a \bmod b$
5. sostituire a con b, b con r ed andare al passo 3



Algoritmo

- Problema: Calcolo del Massimo Comun Divisore tra due numeri a, b :
 $MCD(a, b)$
- Soluzione di Euclide: “ogni divisore comune di a e b è divisore di a , b e del resto r della divisione tra a e b ($a \bmod b$), se questo non è nullo”
- Algoritmo:
 - 1. acquisire due numeri a, b
 - 2. se $b > a$ scambiare a con b
 - 3. se $b = 0$ $MCD(a, b) = a$ a andare al passo 6
 - 4. $r = a \bmod b$
 - 5. sostituire a con b , b con r ed andare al passo 3
 - 6. Fine



Esempio 3

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

```
int main()
```

```
{
```

```
    int a=42;
```

```
    int b=32;
```

```
    int temp;
```

```
    int mcd;
```

```
    int r;
```

```
    if (b>a)
```

```
    {
```

```
        temp=a;
```

```
        a=b;
```

```
        b=temp;
```

```
    }
```

```
    while (b!=0)
```

```
    {
```

```
        r = a % b;
```

```
        a=b;
```

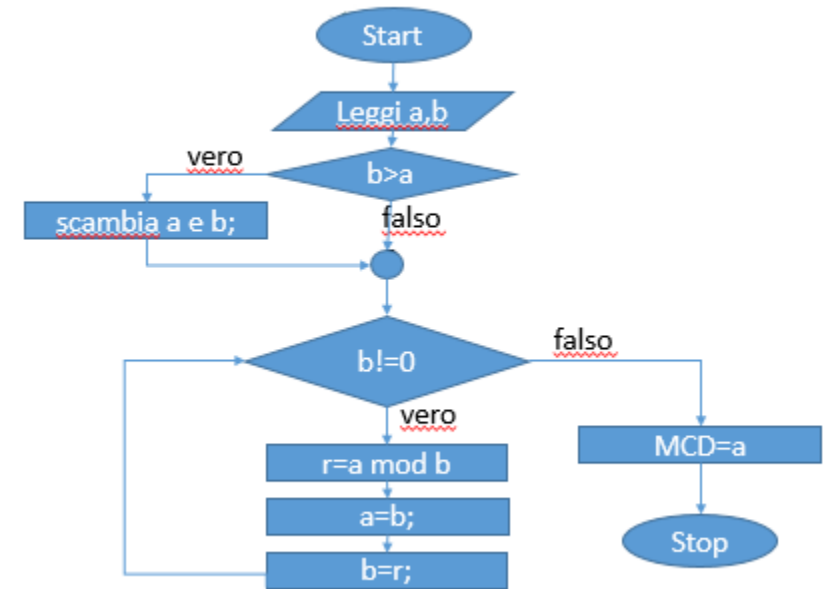
```
        b=r;
```

```
        printf("%d %d %d\n",a,b,r);
```

```
    }
```

```
    mcd=a;
```

```
    printf("mcd = %d",mcd);
```





Input da tastiera

- Sintassi:

```
scanf(<formato>,<arg1>,...,<argn>);
```

< formato>: come per printf

< arg1>,...,<argn>: le variabili cui si vogliono assegnare valori

- **IMPORTANTE:**

I nomi delle variabili vanno precedute dall'operatore & che indica l'indirizzo della variabile (vedremo più avanti il perchè)

- Esempio:

```
int x;
```

```
float z;
```

```
scanf("%d %f", &x, &z);
```



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

- Istruzioni di input vanno viste come assegnazioni dinamiche:
 - L'assegnazione dei valori alle variabili avviene al tempo di esecuzione e viene deciso dall'utente
 - Assegnazioni tradizionali = Assegnazioni statiche
 - L'assegnazione dei valori alle variabili è scritta nel codice!



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Gestire preconditione e postcondizioni (1)

```
if ((a > 0) && (b > 0))
```

```
{
```

```
[...]
```

```
}
```

```
else
```

```
    printf("MCD definito solo per a > 0 e b > 0");
```




Esempio 3

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

```
int main()
```

```
{
```

```
    int a, b, temp;
```

```
    int mcd, r;
```

```
    printf("inserisci a: ");
```

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

```
    printf("inserisci b: ");
```

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

```
    if ((a > 0) && (b > 0))  
    {
```

```
        if (b>a)  
        {
```

```
            temp=a;
```

```
            a=b;
```

```
            b=temp;
```

```
        }
```

```
        while (b!=0)
```

```
        {
```

```
            r = a % b;
```

```
            a=b;
```

```
            b=r;
```

```
            printf("%d %d %d\n",a,b,r);
```

```
        }
```

```
    mcd=a;
```

```
        printf("mcd = %d\n",mcd);  
    }
```

```
    else
```

```
        printf("MCD definito solo per a > 0 e b > 0");  
    }
```

MCD CON INPUT DA TASTIERA



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Gestire precondizione e postcondizioni (2)

```
a = 0;
b = 0;
//while (!(a > 0)&&(b > 0))
while ((a <= 0) || (b <=0))
{
    printf("inserisci a: ");
    scanf("%d",&a);

    printf("inserisci b: ");
    scanf("%d",&b);

    if((a <= 0) || (b <=0))
        printf("MCD definito solo per a > 0 e b >0, reinserire i valori\n");
}
```



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Il costrutto do-while

```
do {  
    Istruzione  
    Istruzione  
    Istruzione  
    Istruzione  
}while(condizione)
```

- Costrutto di ciclo a condizione finale
- Ripete almeno 1 volta il ciclo



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Gestire precondizione e postcondizioni (2)

```
do {  
    printf("inserisci a: ");  
    scanf("%d",&a);  
  
    printf("inserisci b: ");  
    scanf("%d",&b);  
  
    if((a <= 0) || (b <=0))  
        printf("MCD definito solo per a > 0 e b >0, reinserire i valori\n");  
}while ((a <= 0) || (b <=0));
```



Esempio 3

```
#include <stdio.h>

int main()
{
    int a, b, temp;
    int mcd, r;
    a = 0;
    b = 0;
    //while (!((a > 0)&&(b > 0)))
    while ((a <= 0) || (b <=0))
    {
        printf("inserisci a: ");
        scanf("%d",&a);
        printf("inserisci b: ");
        scanf("%d",&b);
```

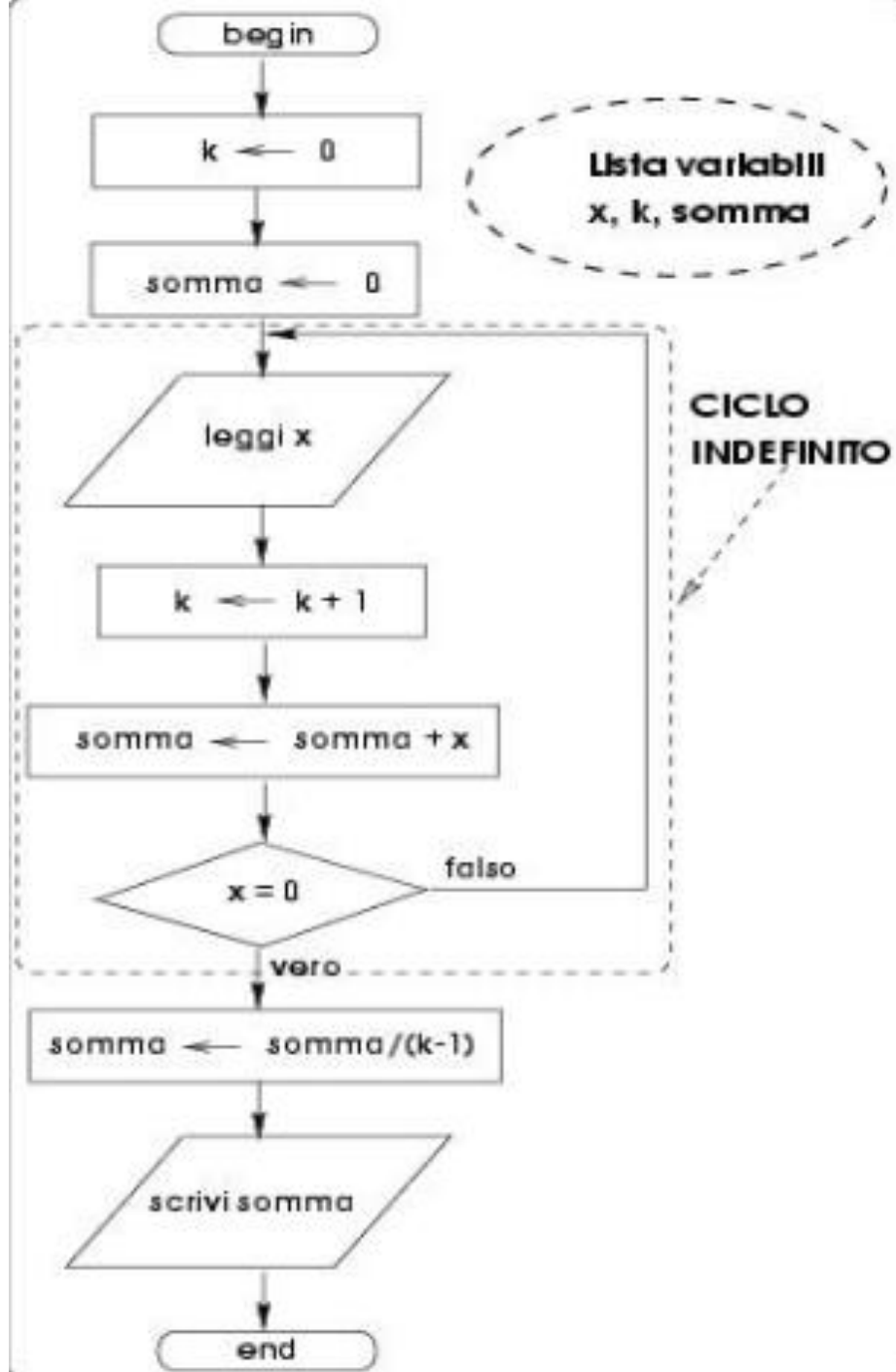
```
        if((a <= 0) || (b <=0))
            printf("MCD definito solo per a > 0 e b >0, reinserire i valori\n");
        }

        if ((a > 0) && (b >0))
        {
            if (b>a)
            {
                temp=a;
                a=b;
                b=temp;
            }
        }
    }
}
```

```
while (b!=0)
{
    r = a % b;
    a=b;
    b=r;
    printf("%d %d %d\n",a,b,r);
}
mcd=a;
printf("mcd = %d",mcd);
}
}
```

MCD CON PRECONDIZIONI

Esempio 4



- Calcolo della media di un insieme di numeri; non è noto a priori quanti sono i numeri di cui deve essere calcolata la media.
- I numeri vengono letti uno alla volta fino a che non si incontra un $x=0$, che segnala la fine dell'insieme



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 4

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

```
int main()
```

```
{
```

```
    int n;
```

```
    float a, media;
```

```
    n = 0;
```

```
    media = 0;
```

```
    do
```

```
    {
```

```
        printf("inserisci un numero (0 per terminare): ");
```

```
        scanf("%f",&a);
```

```
        if(a!=0)
```

```
        {
```

```
            n=n+1; //n++;
```

```
            media = media +a;
```

```
        }
```

```
        }while(a!=0);
```

```
    if(n>0){
```

```
        media = media/n;
```

```
        printf ("media= %f\n",media);
```

```
    }
```

```
}
```