



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](mailto: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

# AVVISO:

Giovedì 30/03 e giovedì 06/04 la lezione si svolgerà nel laboratorio di Informatica



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

## **ESERCITAZIONE**

*Lucidi tratti da: Esercizi di programmazione in C  
Esercitazioni per il corso di Fondamenti di Informatica  
Fulvio Corno Silvia Chiusano  
Politecnico di Torino – Dipartimento di Automatica e  
Informatica*



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

# ***Esempio 1***

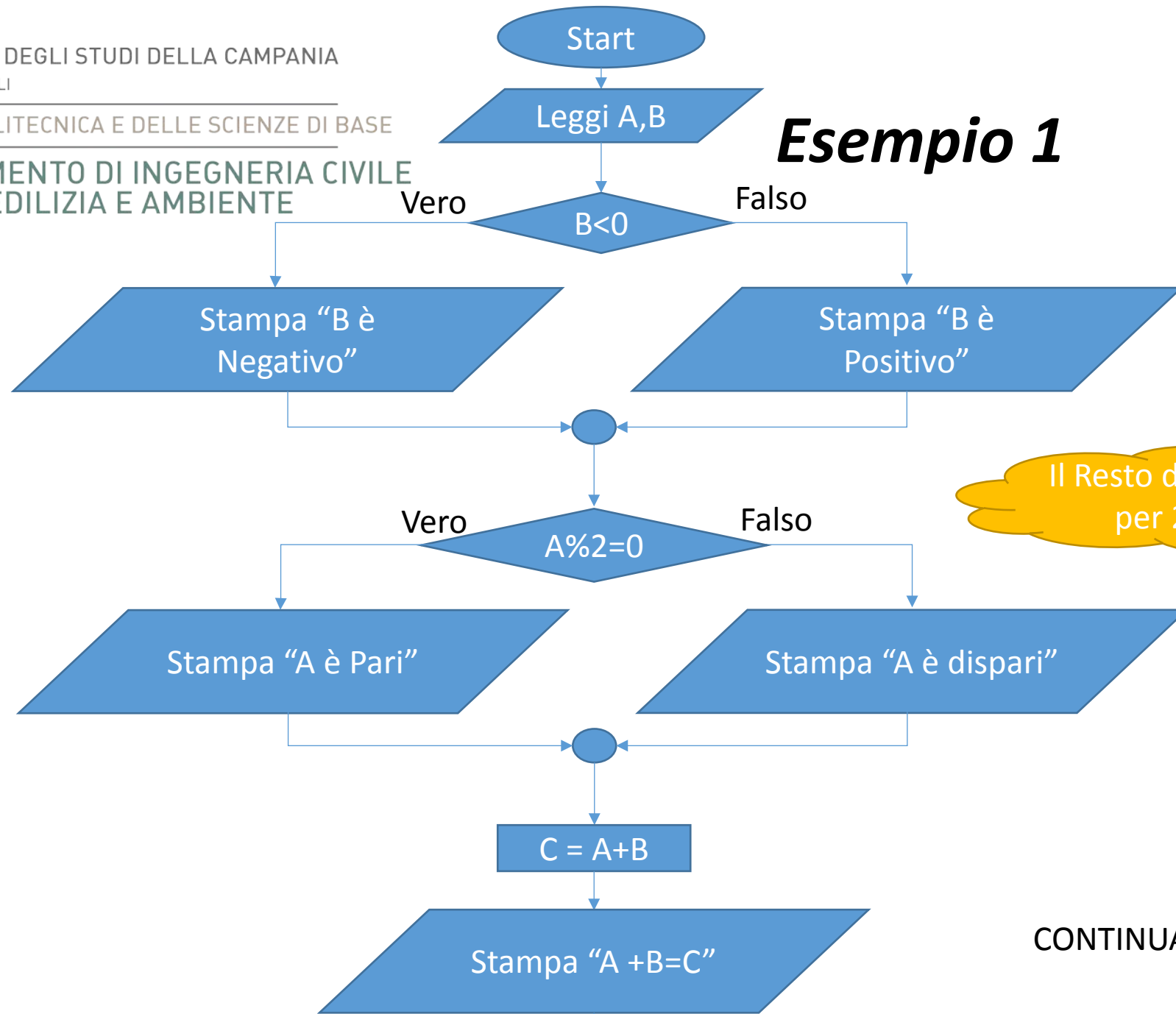
***Si scriva un programma in linguaggio C che legga due numeri da tastiera, detti A e B, e determini le seguenti informazioni, stampandole a video:***

- 1. determini se B è un numero positivo o negativo***
- 2. determini se A è un numero pari o dispari***
- 3. calcoli il valore di  $A + B$***
- 4. determini quale scelta dei segni nell'espressione  $(+A) + (+B)$  porta al risultato massimo, e quale è questo valore massimo.***

***Suggerimento. Nel punto 4., il valore massimo della somma di A e B si può ottenere sommando il valore assoluto di A e di B.***



# Esempio 1

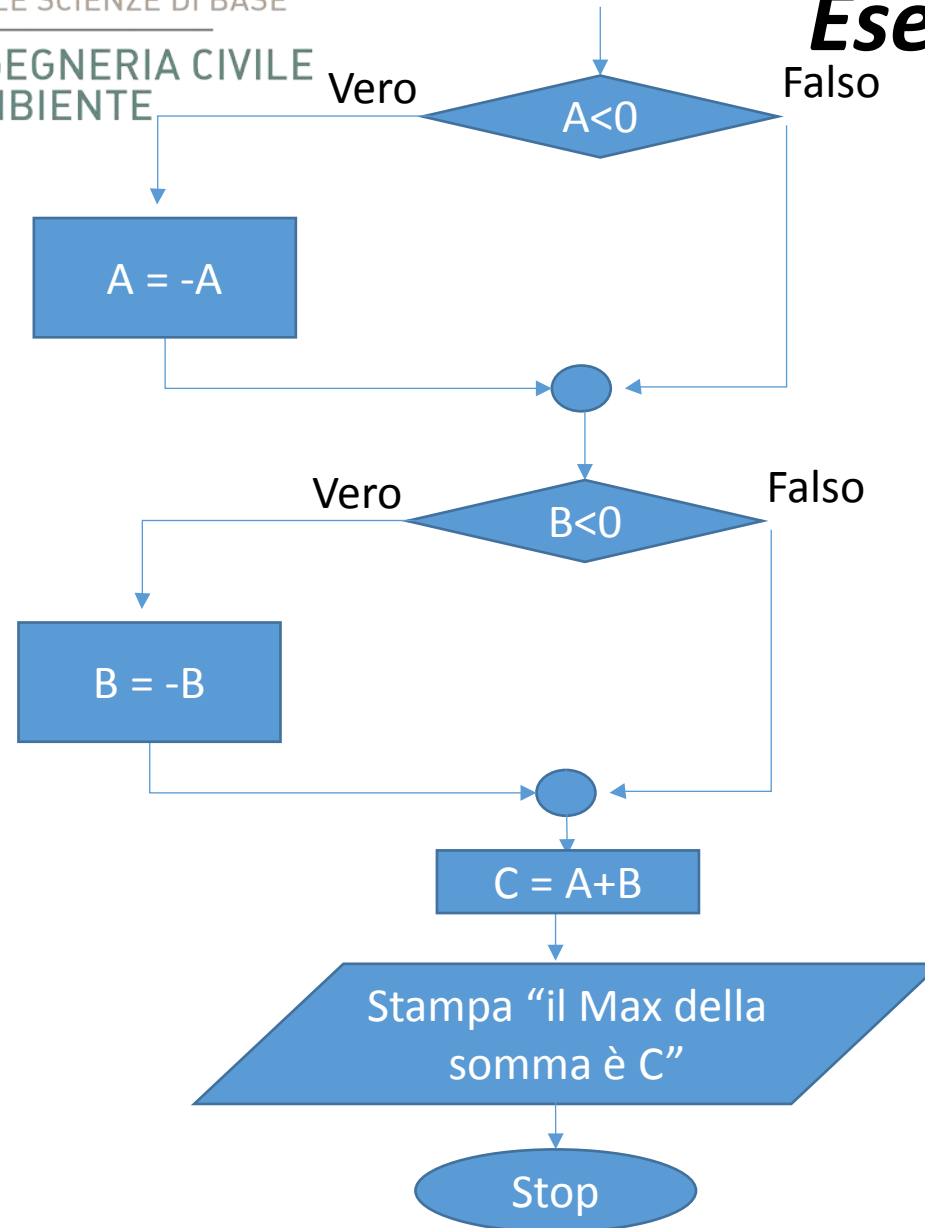


Il Resto della divisione per 2 è zero?

CONTINUA...



# Esempio 1



```

/* PROGRAMMAZIONE IN C */
/* File: controlla-ab-v2.c */
/* Soluzione proposta esercizio "Controlla A e B" */
#include <stdio.h>
int main()
{
int a, b ; /* numeri inseriti A e B */
int c ; /* somma A+B */
/* LEGGI A e B */
printf("Immetti A: ");
scanf("%d", &a);
printf("Immetti B: ");
scanf("%d", &b);
/* CONTROLLA IL SEGNO DI B E STAMPA IL MESSAGGIO
OPPORTUNO */
if( b >= 0 )
{printf("B e' positivo\n");;}
else
{printf("B e' negativo\n");;}
/* CONTROLLA LA PARITA' DI A E STAMPA IL MESSAGGIO
OPPORTUNO */
/* A e' pari se il resto della divisione di A per 2 e' uguale a zero */

```

```

if( a%2 == 0 )
{printf("A e' pari\n");;}
else
{printf("A e' dispari\n");;}
/* CALCOLA A+B E STAMPA IL RISULTATO */
c = a + b ;
printf("La somma %d + %d e' uguale a %d\n", a, b, c);
/* CALCOLA IL VALORE MASSIMO DELLA SOMMA (+- A)
+ (+- B) E STAMPA IL RISULTATO*/
/* Il valore massimo è ottenuto sommando il valore
assoluto di A e di B */
/* Calcola il valore assoluto di A */
if( a < 0 )
a = -a ;
/* Calcola il valore assoluto di B */
if( b < 0 )
b = -b ;
printf("Il valore massimo della somma +-A + +-B e'
uguale a %d\n", a+b );
}

```



## ***Esempio 2***

***Si scriva un programma in linguaggio C che legga da tastiera i valori delle lunghezze dei tre lati di un triangolo (detti A, B e C), e determini:***

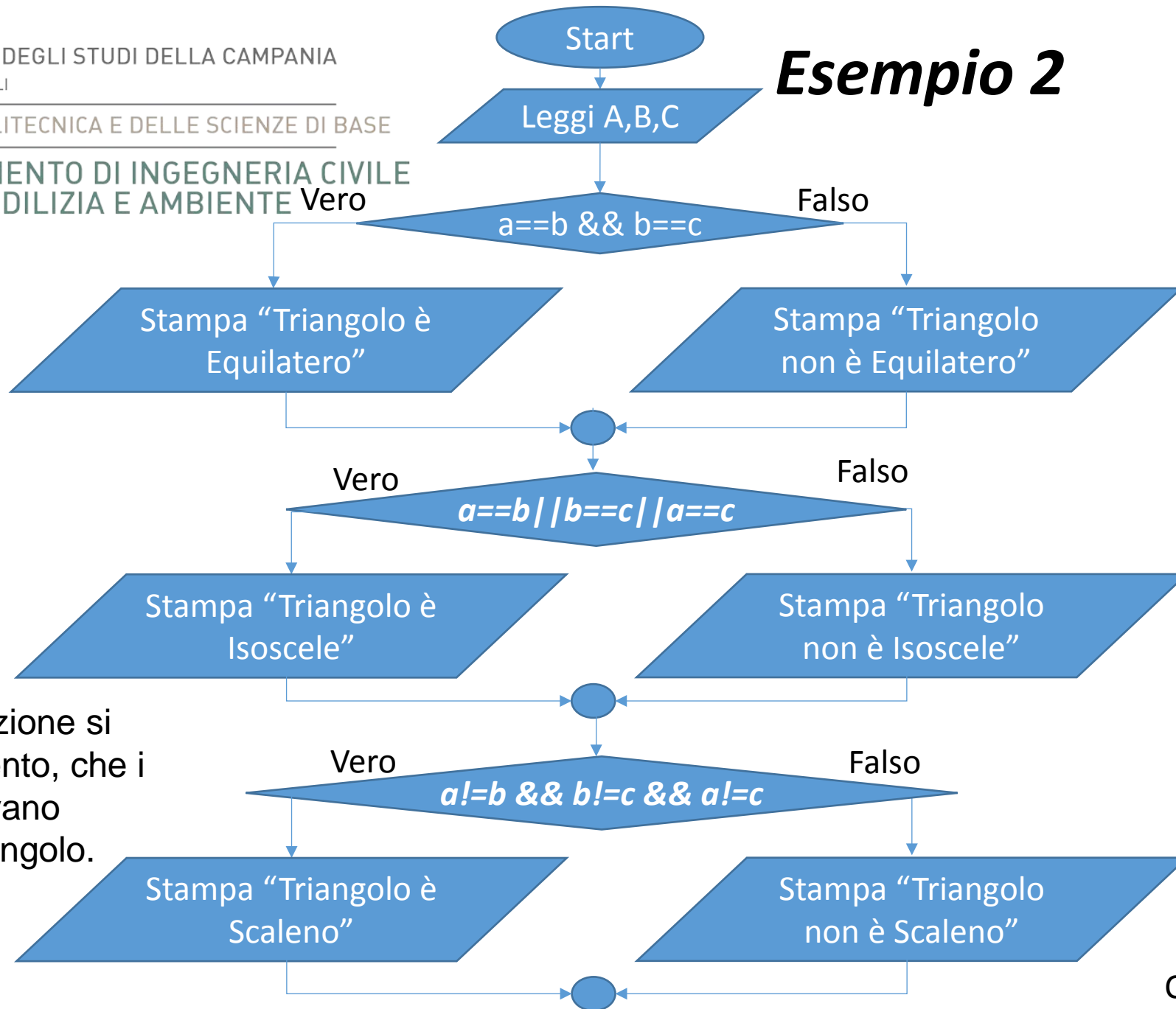
- ***se il triangolo è equilatero***
- ***se il triangolo è isoscele***
- ***se il triangolo è scaleno***
- ***se il triangolo è rettangolo***

In questa prima soluzione si assume, per il momento, che i valori A, B, C descrivano correttamente un triangolo.





# Esempio 2

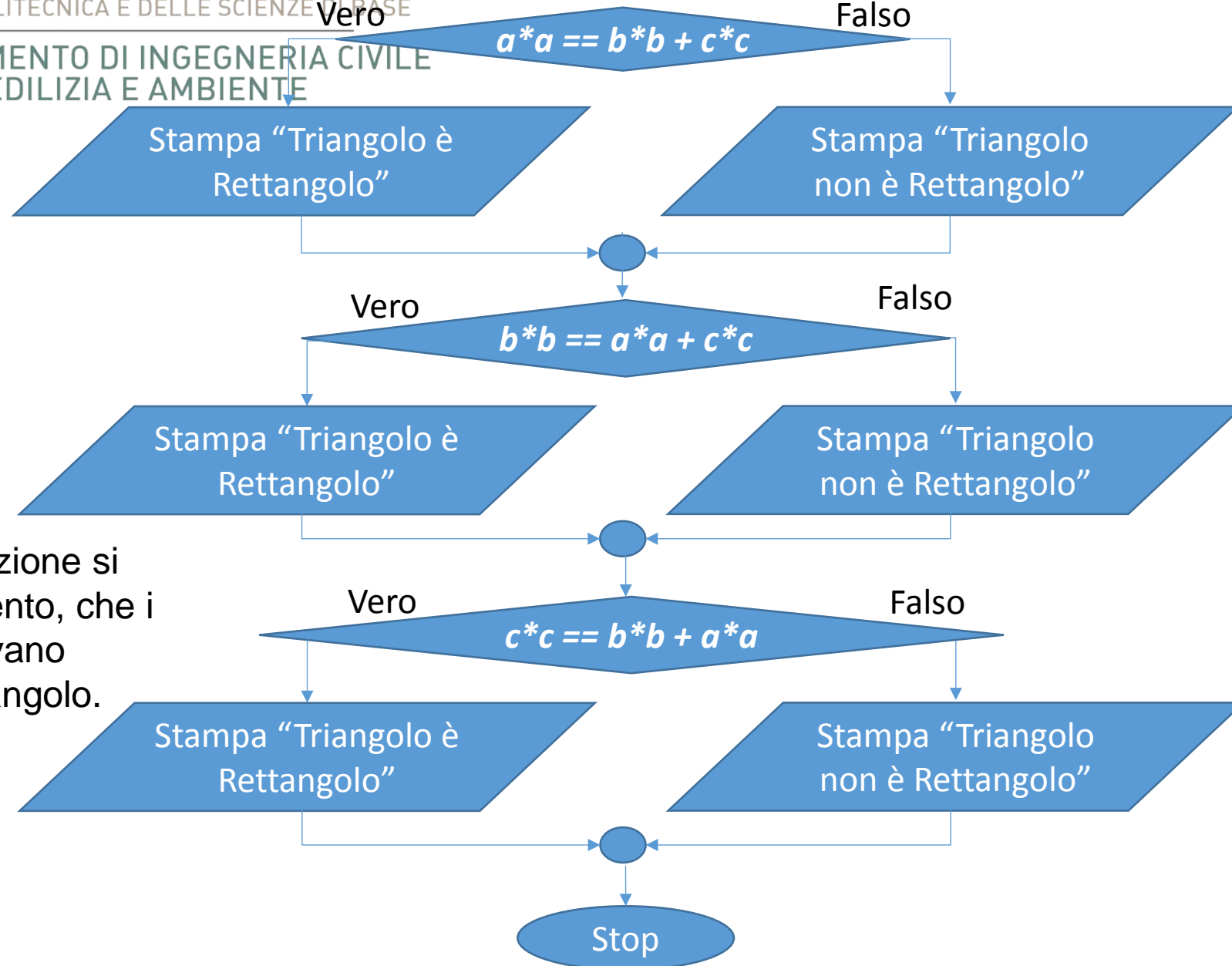


In questa prima soluzione si assume, per il momento, che i valori A, B, C descrivano correttamente un triangolo.

CONTINUA...



# Esempio 2



In questa prima soluzione si assume, per il momento, che i valori A, B, C descrivano correttamente un triangolo.



## Esempio 2

```
/* PROGRAMMAZIONE IN C */
```

```
/* File: triangolo-v1.c */
```

```
/* Soluzione proposta esercizio "Classificazione  
triangolo" (soluzione parziale) */
```

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
float a, b, c; /* lati del triangolo */
```

```
/* LEGGI A, B e C */
```

```
printf("Immetti A: ");
```

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

```
printf("Immetti B: ");
```

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

```
printf("Immetti C: ");
```

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

```
printf("Verifico le proprieta' del triangolo di lati: %f, %f, %f\n", a, b, c);
```

```
/* VERIFICA SE E' EQUILATERO (3 LATI UGUALI)*/
```

```
if( a==b && b==c )
```

```
printf("Il triangolo e' equilatero\n");
```

```
else
```

```
printf("Il triangolo non e' equilatero\n");
```

```
/* VERIFICA SE E' ISOSCELE (2 LATI UGUALI)*/
```

```
if( a==b || b==c || a==c )
```

```
printf("Il triangolo e' isoscele\n");
```

```
else
```

```
printf("Il triangolo non e' isoscele\n");
```

```
/* VERIFICA SE E' SCALENO (3 LATI DIVERSI)*/
```

```
if( a!=b && b!=c && a!=c )
```

```
printf("Il triangolo e' scaleno\n");
```

```
else
```

```
printf("Il triangolo non e' scaleno\n");
```



## Esempio 2

```
/* VERIFICA SE E' RETTANGOLO (TEOREMA DI PITAGORA) */  
/* verifica se il lato A e' l'ipotenusa */  
if(  $a*a == b*b + c*c$  )  
printf("Il triangolo e' rettangolo (ipotenusa A)\n");  
else  
printf("Il triangolo non e' rettangolo (ipotenusa A)\n");  
/* verifica se il lato B e' l'ipotenusa */  
if (  $b*b == a*a + c*c$  )  
printf("Il triangolo e' rettangolo (ipotenusa B)\n");  
else  
printf("Il triangolo non e' rettangolo (ipotenusa B)\n");  
/* verifica se il lato C e' l'ipotenusa */  
if(  $c*c == b*b + a*a$  )
```

```
printf("Il triangolo e' rettangolo (ipotenusa C)\n");  
else  
printf("Il triangolo non e' rettangolo (ipotenusa C)\n");  
/* verifica se il triangolo e' rettangolo */  
if ( (  $a*a == b*b + c*c$  ) ||  
      (  $b*b == a*a + c*c$  ) ||  
      (  $c*c == b*b + a*a$  ) )  
printf("Il triangolo e' rettangolo\n");  
else  
printf("Il triangolo non e' rettangolo\n");  
}
```



## Esempio 3

### Data l'equazione

$$ax + b = 0$$

con  $a$  e  $b$  inseriti da tastiera, scrivere un programma in linguaggio C per determinare il valore di  $x$ , se esiste, che risolve l'equazione.

$$ax + b = 0$$

La soluzione dipende dai valori delle costanti  $a$  e  $b$ :

· se  $a = 0$  e  $b \neq 0$  l'equazione non ha soluzione e si dice *impossibile*  
e di conseguenza nella (\*) membro di sinistra  $\neq$  membro di destra

· se  $a = b = 0$  l'equazione è soddisfatta per qualsiasi valore della variabile e si dice *indeterminata*  
e di conseguenza nella (\*) membro di sinistra  $\equiv$  membro di destra

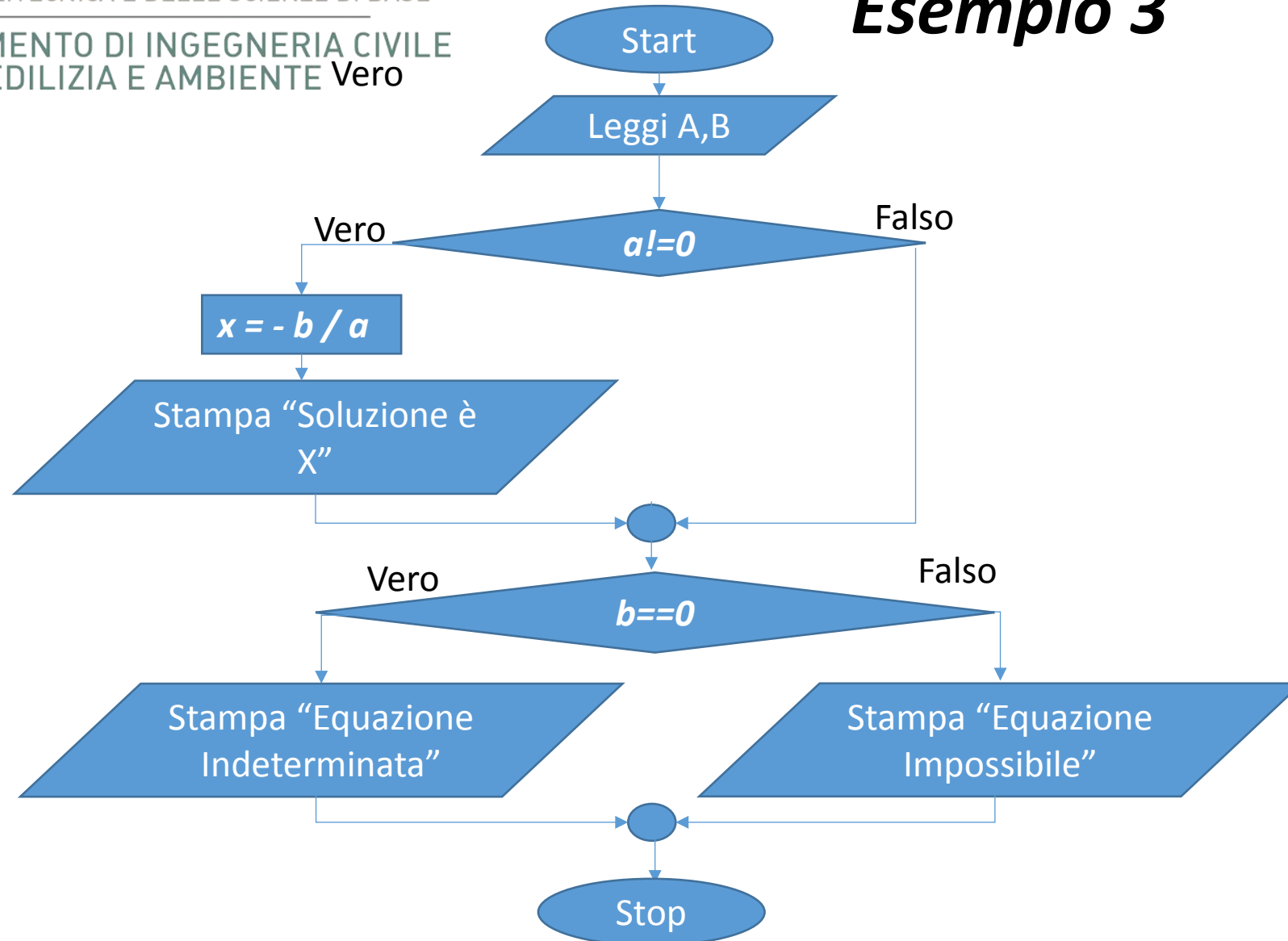
· se  $a \neq 0$  l'equazione si dice determinata ed ha una e una sola soluzione

$$x = - b/a$$

e di conseguenza nella (\*) membro di sinistra = membro di destra



# Esempio 3





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

```
/* PROGRAMMAZIONE IN C */
```

```
/* File: primogrado.c */
```

```
/* Soluzione proposta esercizio "Equazione di  
primo grado" */
```

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

```
int main()
```

```
{
```

```
float a, b ; /* coefficienti a e b */
```

```
float x ; /* valore di x che risolve l'equazione */
```

```
printf("Risoluzione equazioni di primo grado\n");
```

```
printf("Equazione nella forma: ax + b = 0\n");
```

```
/* LEGGI a e b */
```

```
printf("Immetti coefficiente a: ");
```

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

```
printf("Immetti coefficiente b: ");
```

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

```
/* x VIENE CALCOLATO COME  $x = -b/a$ . SI DEVONO  
VERIFICARE I VALORI DI a E b */
```

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

```
{
```

```
x = - b / a ;
```

```
printf("La soluzione e' x = %f\n", x) ;
```

```
}
```

```
else
```

```
{
```

```
/* CASO a==0 */
```

```
if( b==0 )
```

```
{
```

```
printf("Equazione indeterminata (ammette infinite  
soluzioni)\n");
```

```
}
```

```
else
```

```
{
```

```
printf("Equazione impossibile (non ammette soluzioni)\n");
```

```
}
```

```
}
```

```
}
```



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

## ***Esempio 4***

***Si realizzi un programma in linguaggio C per risolvere equazioni di secondo grado. In particolare, data una generica equazione di secondo grado nella forma***

$$ax^2 + bx + c = 0$$

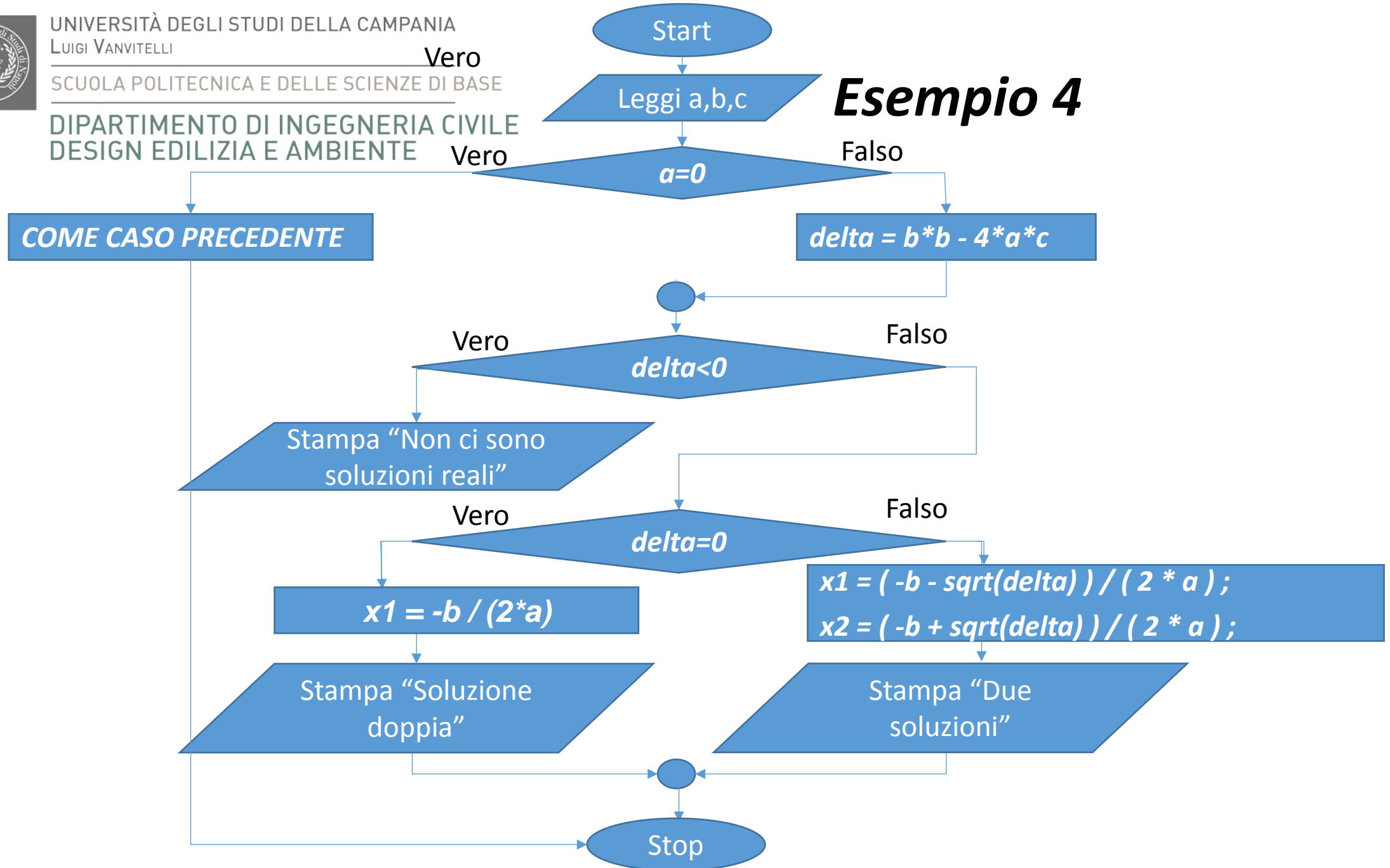
***dove  $a$ ,  $b$ ,  $c$  sono coefficienti reali noti e  $x$  rappresenta l'incognita, il programma determini le due radici  $x_1$  ed  $x_2$  dell'equazione data, ove esse esistano.***

***Si identifichino tutti i casi particolari e si stampino gli opportuni messaggi informativi.***





# Esempio 4





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

```
/* PROGRAMMAZIONE IN C */  
/* File: secondogrado.c */  
/* Soluzione proposta esercizio "Equazione di secondo grado" */  
#include <stdio.h>  
#include <math.h>  
int main()  
{  
float a, b, c ; /* coefficienti a, b e c */  
float delta ; /* discriminante */  
float x1, x2 ; /* valori di x che risolvono l'equazione */  
printf("Risoluzione equazioni di secondo grado\n");  
printf("Equazione nella forma: ax^2 + bx+ c = 0\n");  
/* LEGGI a e b */  
printf("Immetti coefficiente a: ");  
scanf("%f", &a);  
printf("Immetti coefficiente b: ");  
scanf("%f", &b);  
printf("Immetti coefficiente c: ");  
scanf("%f", &c);
```

```
/* Se a==0, allora in realtà è un'equazione di primo grado */  
if ( a==0 )  
{ /* x VIENE CALCOLATO COME x=-c/b.SI DEVONO VERIFICARE I  
VALORI DI b E c */  
if ( b != 0 )  
{ x1 = - c / b ; printf("Una soluzione: x = %f\n", x1) ; }  
else { /* CASO b==0 */  
if ( b==0 )  
{ printf("Equazione indeterminata (ammette infinite soluzioni)\n"); }  
else { printf("Equazione impossibile (non ammette soluzioni)\n"); } }  
} else /* a != 0, quindi è una 'vera' equazione di secondo grado */  
{ /* Calcoliamo il discriminante 'delta' */  
delta = b*b - 4*a*c ;  
if ( delta<0 ) { printf("Non ci sono soluzioni in campo reale\n") ; }  
else if ( delta == 0 )  
{ x1 = -b / (2*a) ; printf("Una soluzione doppia: x = %f\n", x1) ;  
} else /* delta > 0 */  
{ /* caso normale */  
x1 = ( -b - sqrt(delta) ) / ( 2 * a ) ;  
x2 = ( -b + sqrt(delta) ) / ( 2 * a ) ;  
printf("Due soluzioni: x = %f e x = %f\n", x1, x2 ) ; }  
}}
```



## Esempio 5

***Determinare il valore visualizzato dai seguenti programma nel caso in cui num=4 e per i seguenti valori della variabile conta: conta=5, conta=0, conta=1, conta= -5.***

```
int conta, num;
scanf("%d", &conta);
scanf("%d", &num);

while (conta != 0)
{
    num = num * 10;
    conta = conta - 1;
}
printf("%d\n", num) ;
```

```
int conta, num;
scanf("%d", &conta);
scanf("%d", &num);

while (conta > 0)
{
    num = num * 10;
    conta = conta - 1;
}
printf("%d\n", num) ;
```



## Esempio 5

Si nota innanzitutto come i due programmi siano identici tranne per la condizione  $\text{conta} \neq 0$ , che diviene  $\text{conta} > 0$  nel secondo. Ciò significa che i due programmi si comporteranno in modo identico ogniqualvolta  $\text{conta}$  sarà un valore positivo o nullo (perché in tal caso le due condizioni  $\text{conta} \neq 0$  e  $\text{conta} > 0$  si equivalgono), mentre si potranno comportare diversamente quando  $\text{conta} < 0$ .

Analizzando il ciclo, si nota che l'operazione principale eseguita è  $\text{num} = \text{num} * 10$ , che viene ripetuta  $\text{conta}$  volte. In pratica il programma calcola un valore finale pari a  $\text{num} \cdot 10^{\text{conta}}$ .

In definitiva il valore calcolato (e stampato) sarà:

	Programma di sinistra	Programma di destra
<code>num=4, conta=5</code>	400000	400000
<code>num=4, conta=0</code>	4	4
<code>num=4, conta=1</code>	40	40
<code>num=4, conta=-5</code>	(*)	4

(\*) in questo caso il programma esibisce un comportamento anomalo, dovuto ad un errore di programmazione (non ci si è "protetti" contro un dato errato, ossia negativo, inserito dall'utente).

Il ciclo viene eseguito un'enormità di volte (dell'ordine di 232 volte), finché il valore di  $\text{conta}$ , che parte da -5 e viene decrementato ripetutamente fino a quando la sottrazione non andrà in overflow, e poi nuovamente finché non arriverà a zero. In tal caso  $\text{num}$  viene moltiplicato per 10 un'enormità di volte, andando ripetutamente in overflow... il risultato ottenuto sarà quindi totalmente imprevedibile (e tra l'altro dipendente dall'implementazione degli `int` nel compilatore utilizzato). A titolo di esempio, nel caso del compilatore Dev-C++ su piattaforma Windows, dopo circa 20 secondi (durante i quali il programma decrementa  $\text{conta}$  all'impazzata) viene stampato il valore 0.



## ***Esempio 6***

***Si scriva un programma in linguaggio C che converta un numero binario in un numero decimale. Il numero binario è rappresentato su  $N$  bit, e il valore di  $N$  è inserito da tastiera.***

***L'utente inserisce le cifre del numero binario un bit alla volta, partendo dal bit meno significativo (ossia dal bit di peso  $2^0$ ). Il programma visualizzerà il numero decimale corrispondente.***

***Suggerimento. Per calcolare le potenze di 2 utilizzare la funzione `pow`, includendo la libreria `math.h`. Ad esempio per calcolare  $2^5$ , si scriverà `pow(2,5)`. In generale, data una base  $a$ , per calcolare  $y = a^b$ , si scrive `y = pow(a,b)` includendo la libreria `math.h`.***



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

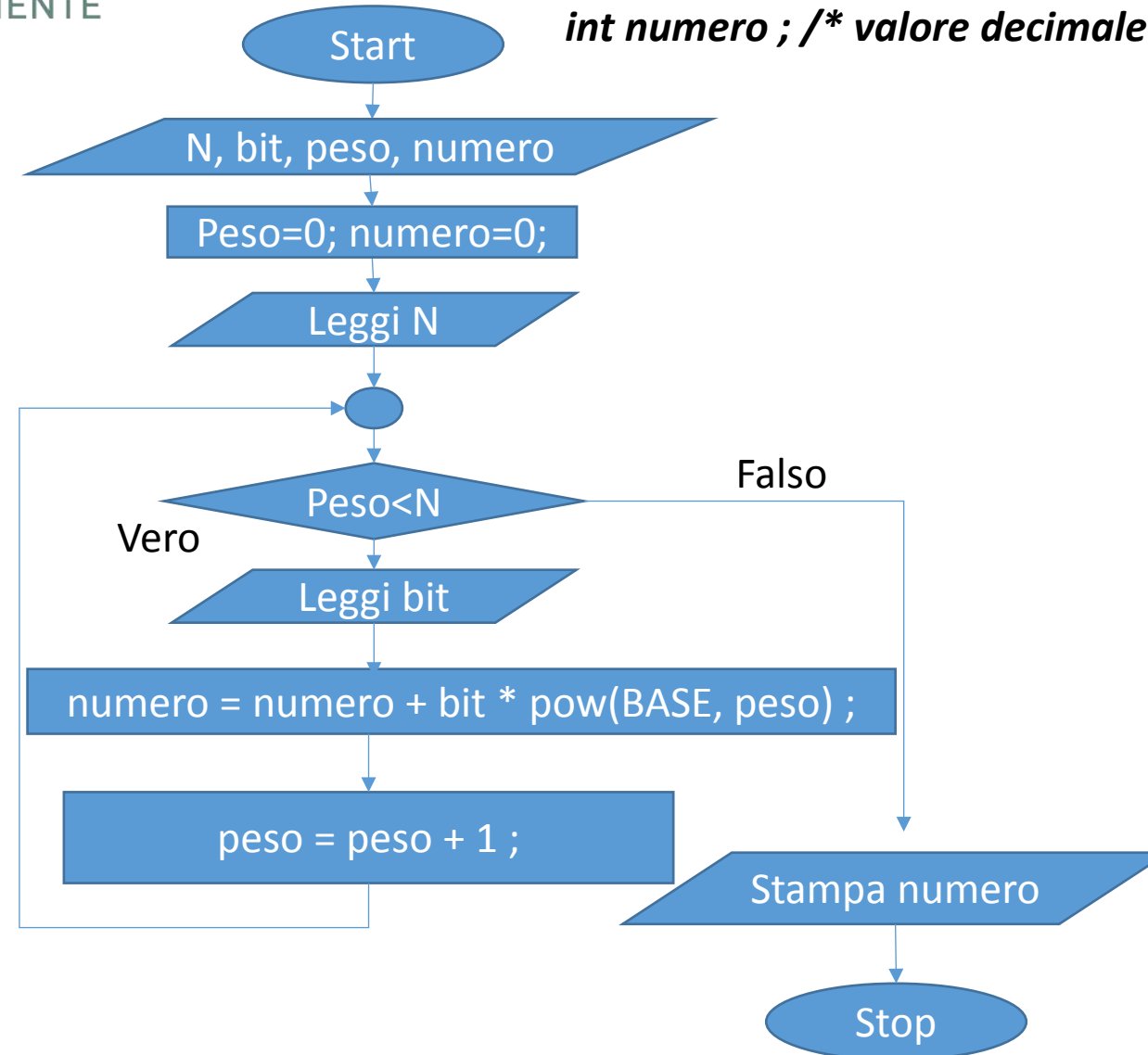
DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

*int N ; /\* numero di cifre binarie \*/*

*int bit ; /\* cifra binaria \*/*

*int peso ; /\* peso della cifra binaria \*/*

*int numero ; /\* valore decimale del numero binario \*/*





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

```
/* PROGRAMMAZIONE IN C */  
/* File: bindec1.c */  
/* Soluzione proposta esercizio "Conversione Binario-Decimale" */  
#include <stdio.h>  
#include <math.h>  
#define BASE 2  
int main()  
{  
int N ; /* numero di cifre binarie */  
int bit ; /* cifra binaria */  
int peso ; /* peso della cifra binaria */  
int numero ; /* valore decimale del numero binario */  
/* INIZIALIZZA LE VARIABILI */  
peso = 0 ; /* LA PRIMA CIFRA BINARIA LETTA AVRA' PESO ZERO */  
numero = 0 ; /* IL VALORE DECIMALE DEL NUMERO BINARIO E'  
INIZIALIZZATO A ZERO */
```

```
/* LEGGI IL NUMERO DI CIFRE BINARIE */  
printf("Immetti il numero di bit del numero binario: ");  
scanf("%d", &N);  
/* LEGGI IL NUMERO BINARIO */  
printf("\nImmetti il numero binario partendo dal bit meno  
significativo \n");  
while ( peso < N )  
{  
/* LEGGI LA CIFRA BINARIA SUCCESSIVA */  
printf("Immetti la cifra binaria 2^%d:", peso);  
scanf("%d", &bit);  
/* CALCOLA IL VALORE IN DECIMALE DELLA CIFRA BINARIA INSERITA  
E AGGIUNGILO ALLA CIFRA DECIMALE CALCOLATA FINO AD ORA */  
numero = numero + bit * pow(BASE, peso);  
/* AGGIORNA IL PESO DELLA CIFRA BINARIA */  
peso = peso + 1;  
}  
/* STAMPA IL RISULTATO */  
printf("\n");  
printf("La cifra decimale calcolata e': %d\n", numero);  
}
```



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

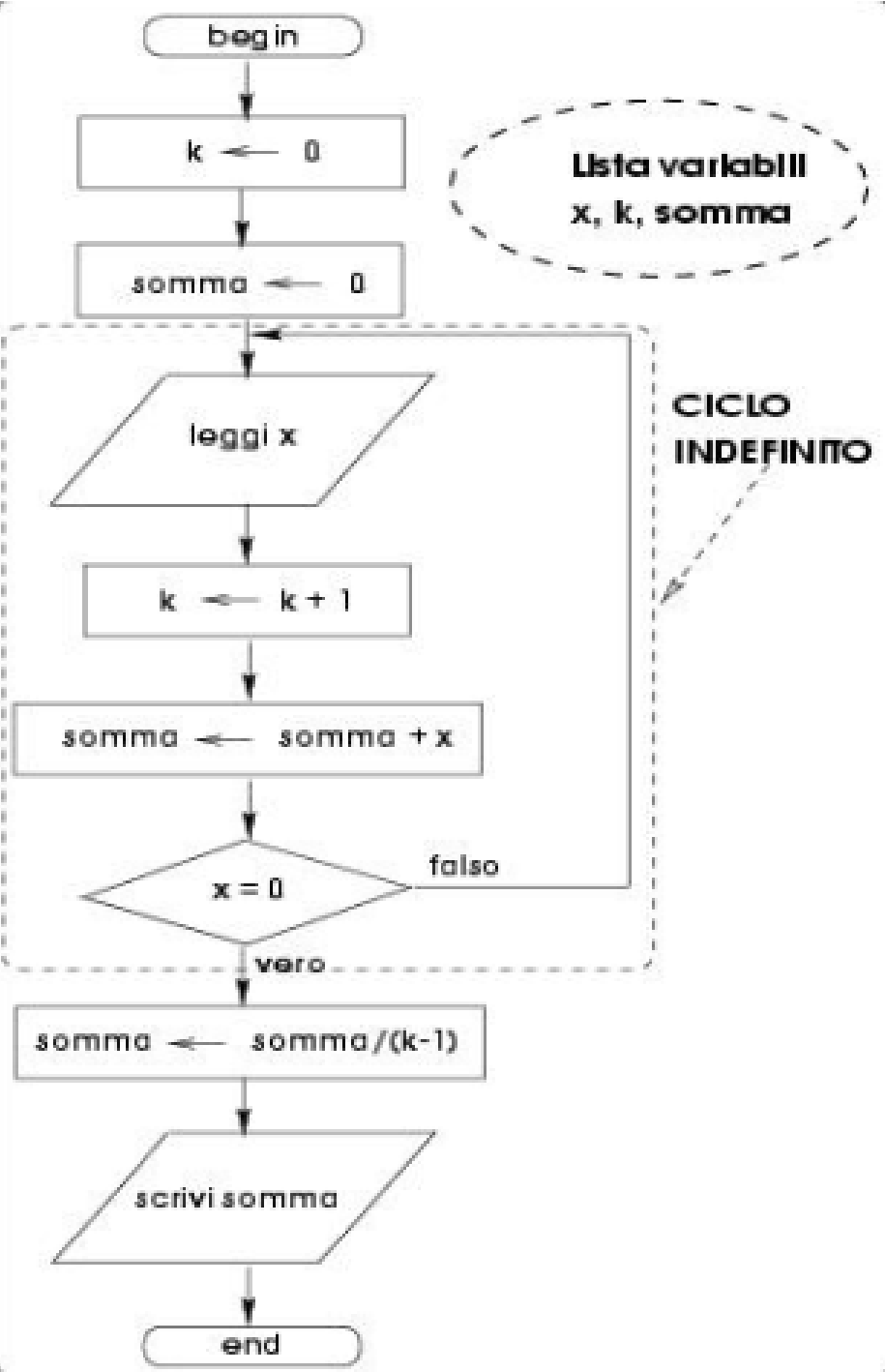
DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

## ***Esempio 7***

***Si scriva un programma in linguaggio C per calcolare la media aritmetica di una serie di numeri inseriti da tastiera. L'introduzione di un valore particolare pari a "0" indica il termine del caricamento dei dati.***



# Esempio 7





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA  
LUIGI VANVITELLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA CIVILE  
DESIGN EDILIZIA E AMBIENTE

```
/* PROGRAMMAZIONE IN C */  
/* File: media_numeri.c */  
/* Soluzione proposta esercizio "Media dei numeri" */  
#include <stdio.h>  
int main()  
{  
int numero ; /* numero inserito */  
int conta ; /* conta quanti numeri sono inseriti */  
float somma ; /* somma dei numeri inseriti */  
float media ; /* media dei numeri inseriti */  
/* "somma" e "media" sono di tipo float per calcolare la media  
come valore decimale con la virgola*/  
/* INIZIALIZZA LE VARIABILI */  
somma = 0 ;  
conta = 0 ;  
/* LEGGI UN NUMERO */  
printf("Inserire una serie di numeri. La condizione di terminazione "  
"e' il numero zero.\n");
```

```
printf("Inserisci numero: ");  
scanf ("%d", &numero) ;  
/* LEGGI UNA SERIE DI NUMERI, FINO A QUANDO NON E' INSERITO IL  
NUMERO 0 */  
while ( numero != 0 )  
{  
/* AGGIORNA LA SOMMA DEI NUMERI INSERITI */  
somma = somma + numero ;  
/* INCREMENTA IL CONTATORE DEI NUMERI INSERITI FINO AD ORA */  
conta = conta + 1 ;  
/* LEGGI UN NUMERO */  
printf("Inserisci numero: ") ;  
scanf ("%d", &numero);}  
/* CALCOLA LA MEDIA DEI NUMERI INSERITI */  
media = somma/conta ;  
/* STAMPA IL RISULTATO */  
printf("\n") ;  
printf("Numeri inseriti %d, Somma %f, Media %f \n", conta, somma,  
media);
```