

Liceo Scientifico e Classico “S. Trinchese”

**ELEMENTI DI
PROGRAMMAZIONE
E TURBO PASCAL**

seminario per la classe IV BS



Prof Antonio Cazzato

Perché programmare?

La programmazione ha una notevole valenza didattica per i seguenti motivi:

1. richiede innanzitutto una buona conoscenza dei processi logici da seguire per ottenere risultati corretti attraverso l'elaborazione dei dati;
2. insegna a formalizzare la soluzione di un problema in modo tale che lo sappia risolvere anche un esecutore di compiti elementari, pertanto consente di gestire la macchina in modo consapevole;
3. per programmare è necessario analizzare molto attentamente il problema da risolvere: se esso è particolarmente complesso lo si scompone in sotto-problemi più semplici; si individuano i dati d'ingresso e d'uscita, i metodi di risoluzione e gli strumenti disponibili.

Tutto ciò aumenta le capacità di astrazione dei giovani allievi.



- Attraverso l'analisi, si passa dal problema all'algoritmo di risoluzione; la programmazione consiste nel tradurre l'algoritmo in programma e, tramite l'elaborazione, si perviene ai risultati.

Le fasi del lavoro, pertanto, sono :

- ANALISI: dal problema all'algoritmo
- PROGRAMMAZIONE: dall'algoritmo al programma
- ELABORAZIONE: attraverso il programma, dai dati di ingresso ai risultati.

ANALISI

Dal problema all'algoritmo

Il processo di risoluzione di un problema può essere diviso in fasi ben distinte tra loro:

- **descrizione del problema**
- **stesura dell'algoritmo.**

Nella prima fase si definisce il problema, descrivendo con precisione i dati in esso coinvolti, quelli a disposizione e i risultati che ci aspettiamo dalla soluzione del problema. Inoltre occorre analizzare quali siano le risorse disponibili.

Nella seconda fase si indicano, in esatta sequenza, le azioni da intraprendere per la soluzione del problema e per il conseguimento dei risultati attesi.

Esempi

- In una ricetta da cucina si indicano dapprima il numero di persone, le dosi e gli ingredienti: questi sono i dati iniziali del problema. Si elencano poi le risorse disponibili ossia gli utensili, i tipi di pentole, il forno e le relative temperature raggiungibili.
Infine si descrive il processo di preparazione della pietanza.
- Nel calcolo dell'area di un rettangolo si indicano dapprima i dati iniziali, ossia le misure della base e dell'altezza e il risultato che si vuole ottenere, ossia il valore dell'area. Si specifica poi lo strumento di calcolo utilizzabile (calcolatrice, elaboratore).
Infine si descrive la sequenza finita di operazioni da eseguire per ottenere l'area partendo dai dati iniziali.

1. Descrizione del problema

- Gli elementi che caratterizzano un problema sono:
 - i dati iniziali anche detti dati d'ingresso o **di input**
 - i risultati che si vogliono ottenere, detti anche dati di uscita o **di output**
 - le risorse a disposizione sia dal punto di vista logico (tabelle, schemi logici, criteri di calcolo) sia dal punto di vista fisico (calcolatrici, elaboratori)
 - le soluzioni adottate, ossia il procedimento che permette di passare dai dati iniziali ai risultati attesi.

2. Stesura dell'algoritmo

La stesura dell'algoritmo consiste nella scomposizione del procedimento risolutivo di un problema in una sequenza finita di operazioni elementari da eseguire per ottenere i risultati attesi.

Del termine algoritmo (dal nome dell'algebrista arabo **Al- Khuwarizmi** vissuto nel IX secolo) occorre dare una definizione precisa, per evidenziare quali siano le sue caratteristiche principali.

3. Definizione di algoritmo

- L'algoritmo è una sequenza finita di azioni elementari che devono essere eseguite per raggiungere un risultato di natura qualsiasi, ma chiaramente e completamente definito.

4. Caratteristiche dell'algoritmo

Un algoritmo per essere corretto deve soddisfare le seguenti proprietà, deve essere:

- **DETERMINISTICO**, ossia ogni successiva esecuzione dello stesso algoritmo con i medesimi dati iniziali deve produrre sempre i medesimi risultati finali;
- **FINITO**, l'esecuzione delle azioni indicate deve terminare in un tempo finito;
- **GENERALE**, ossia completo ed esaustivo nel senso che deve essere indicata la soluzione da seguire per tutti i casi che si possono verificare durante l'esecuzione;
- **PRECISO**, ossia non ambiguo.

5. Algoritmo ed esecutore

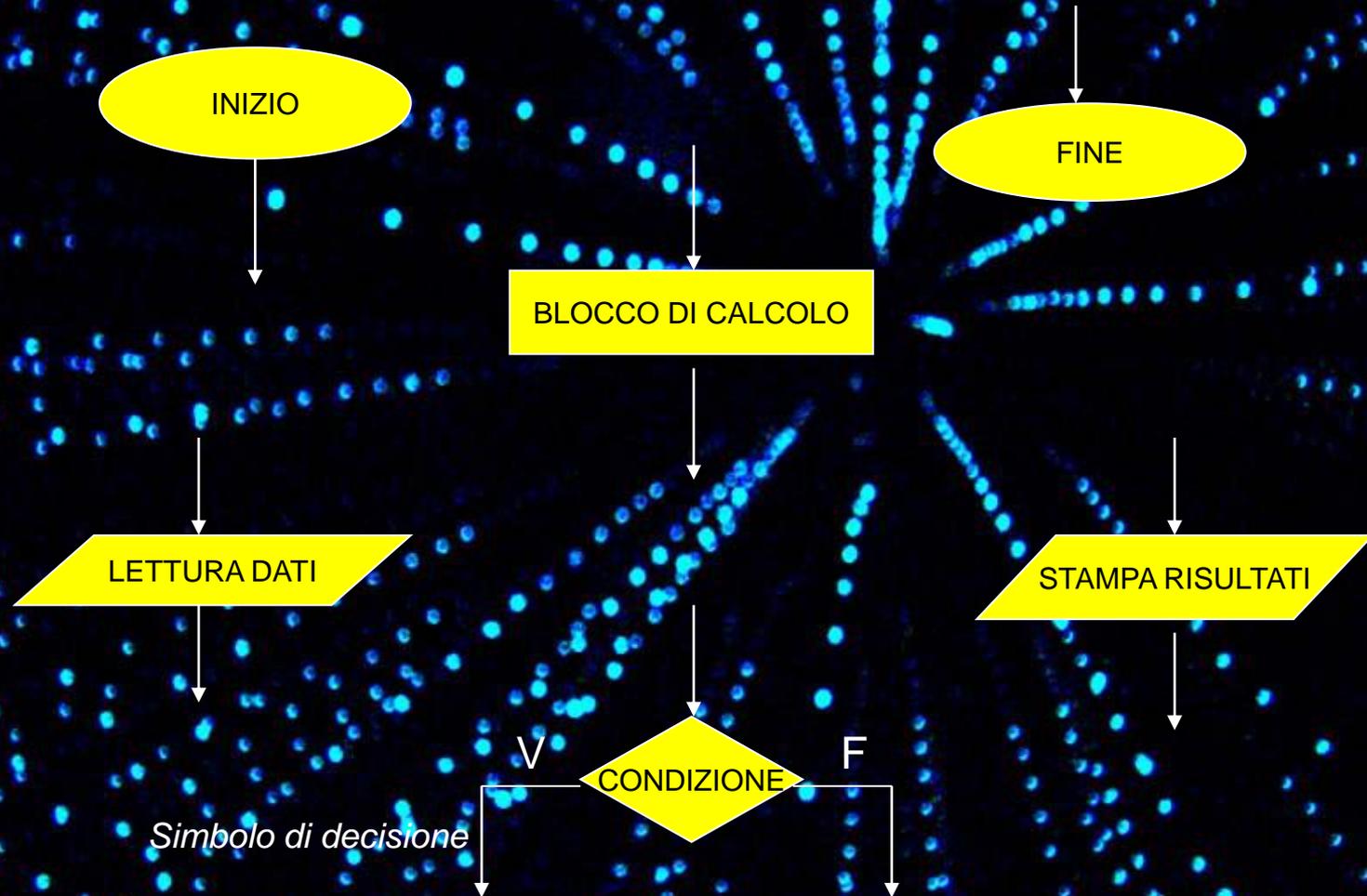
- Per descrivere le azioni è necessario disporre di un linguaggio; la descrizione di un'azione in un linguaggio formale, è detta **istruzione**.
- L'**esecutore** è l'ente che esegue le azioni secondo le istruzioni di un algoritmo. Esso deve essere in grado di acquisire i dati iniziali, attivare il processo di elaborazione e comunicare all'esterno i risultati finali ottenuti.



6. Descrizione dell'algoritmo

- **Diagrammi a blocchi**: descrizione grafica dell'algoritmo attraverso simboli di forma diversa uniti tra loro da linee orientate, che indicano il flusso delle operazioni.
- **Pseudocodifica**: descrizione dell'algoritmo attraverso un linguaggio sintetico, con un vocabolario molto ristretto ed una sintassi molto rigorosa.

Blocchi elementari



La pseudocodifica

- La pseudocodifica utilizza per la descrizione dell'algoritmo termini e parole del linguaggio comune, ma le regole che strutturano il testo vanno seguite con rigore. In pseudocodifica sono utilizzate le **PAROLE CHIAVE** che corrispondono ai passaggi fondamentali dell'algoritmo:

INIZIO e **FINE** aprono e chiudono il testo dell'algoritmo

ESEGUI operazione

RICEVI dati in ingresso

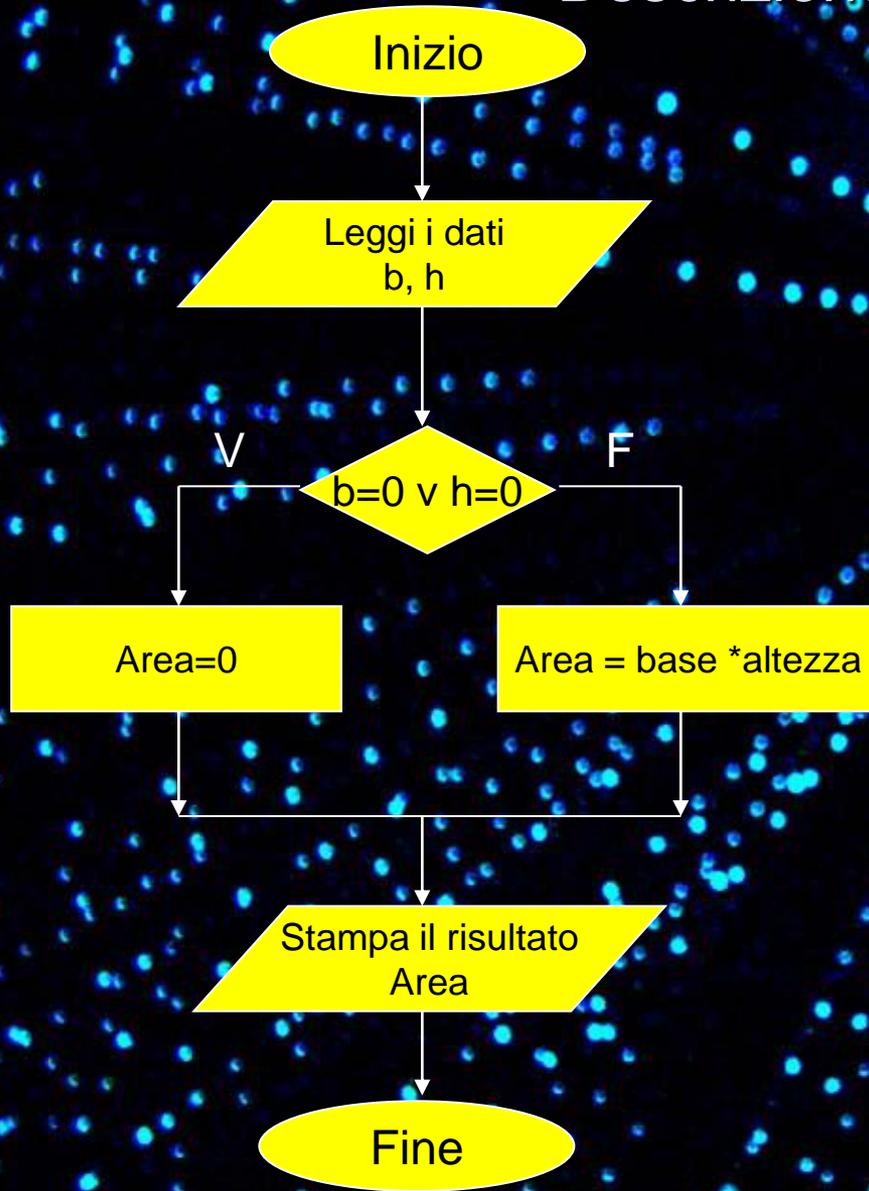
COMUNICA risultato

SE...ALLORA...ALTRIMENTI indicano istruzioni che vengono eseguite in alternativa

ESEGUI... RIPETI FINCHE' indicano istruzioni che devono essere eseguite in ripetizione finchè non risulta verificata una condizione.

PROBLEMA: calcolo dell'area di un rettangolo

Descrizione dell'algoritmo



Inizio programma area del rettangolo

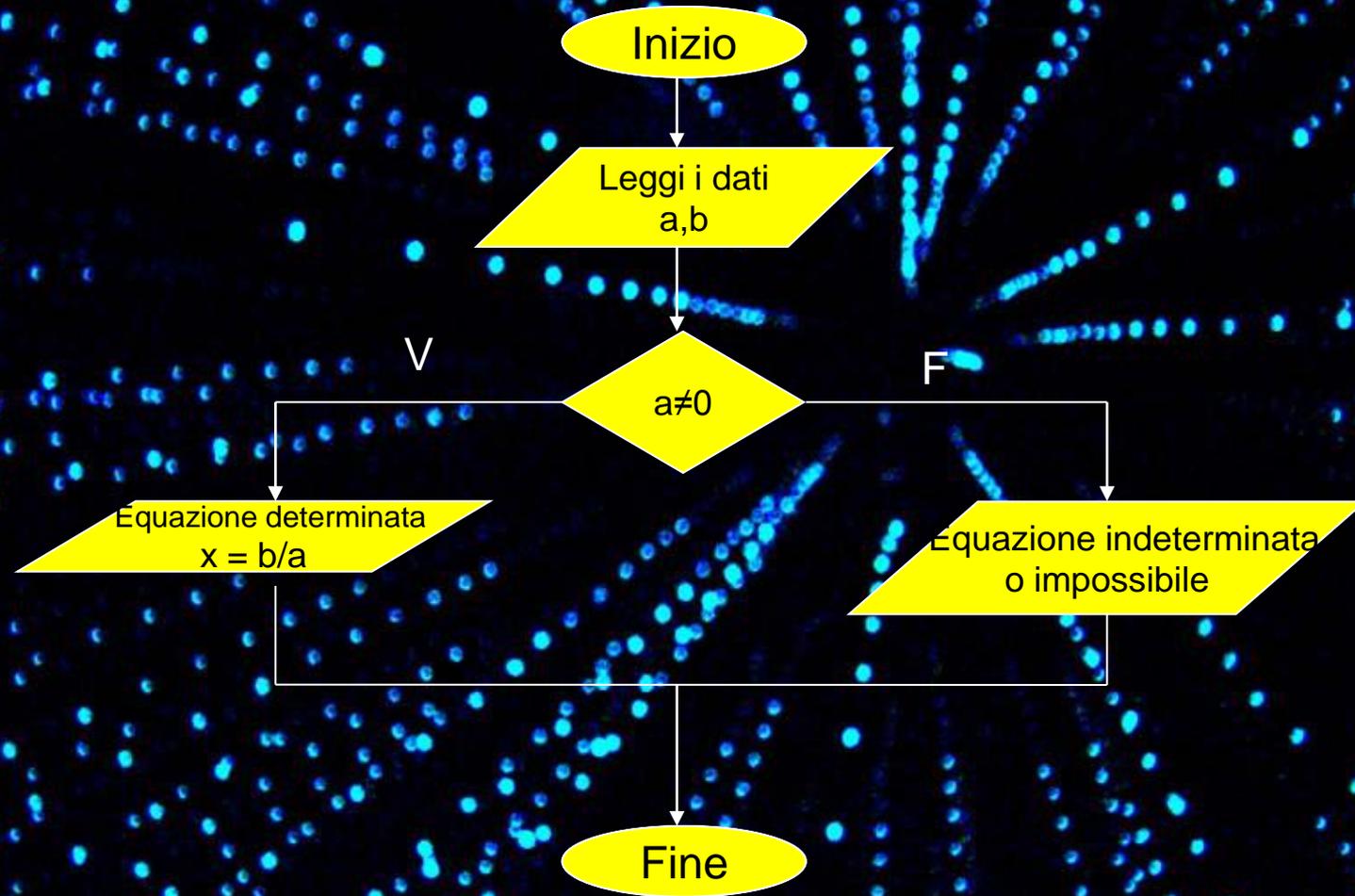
Ricevi base
Ricevi altezza

Se $(b=0) \vee (h=0)$ allora $Area=0$
altrimenti $Area=base * altezza$

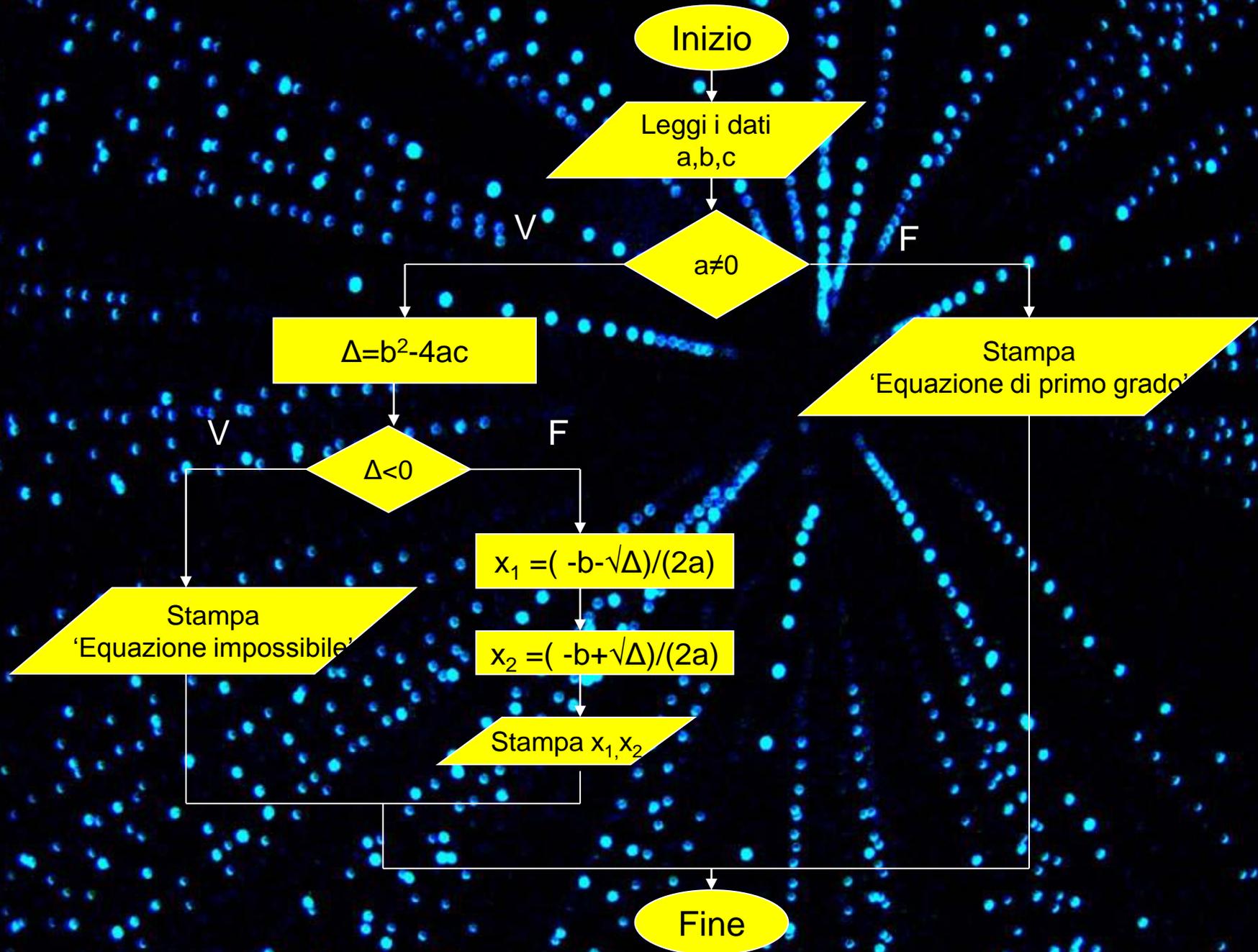
Comunica Area

Fine.

PROBLEMA: risoluzione dell'equazione di primo grado $a \cdot x = b$



Risoluzione dell'equazione di secondo grado $ax^2+bx+c=0$



PROGRAMMAZIONE

Dall'algoritmo al programma

Il **linguaggio di programmazione** è un linguaggio comprensibile per la macchina, attraverso un traduttore.

Il **programmatore** è colui che è in grado di scrivere programmi in linguaggio di programmazione.

Un programma è composto essenzialmente da:

- **istruzioni**, ossia le attività da svolgere;
- **dati**, ossia le entità su cui operano le istruzioni.

Come per i linguaggi naturali, si devono rispettare:

- il **lessico**, cioè l'insieme delle parole riconosciute dal linguaggio;
- la **sintassi**, cioè le modalità per scrivere le frasi in modo corretto.

Il linguaggio Pascal fu ideato dal Niklaus Wirth del politecnico di Zurigo all'inizio degli anni '70. Esso è composto da istruzioni abbastanza vicine al linguaggio comune e per questo motivo è particolarmente adatto a chi affronta per la prima volta l'attività di programmazione.

Calcolo dell'area del rettangolo

Programma in Pascal

```
PROGRAM AreaDelRettangolo;  
USES wincrt;  
VAR b,h,A: real;  
BEGIN  
  WRITELN ('Inserisci il valore della base');  
  READLN (b);  
  WRITELN ('Inserisci il valore dell" altezza');  
  READLN (h);  
  WRITELN ('base = ', b);  
  WRITELN ('altezza = ', h);  
  IF (b=0) OR (h=0) THEN  
    WRITELN ('Area = 0')  
  ELSE  
    BEGIN  
      A := b * h;  
      WRITELN('Area = ', A);  
    END;  
END.
```

PROGRAM nome del programma;

costituisce l'intestazione del programma

con **Uses** si dichiarano i moduli software che si usano per l'esecuzione del programma

Ad esempio:

Uses Printer (per la gestione della stampante)

Uses Wincrt (per la gestione del video)

Nella sezione dichiarativa devono essere indicate le variabili e/o le costanti utilizzate nel programma

con frasi del tipo:

CONST Nome = valore;

VAR Nome: tipo ;

Esempi di dichiarazioni di variabili o di costanti:

CONST Euro = 1936.27;

VAR Eta: Integer ; (interi relativi [-32768...32767])

Altezza: **real** ;

Risposta: **char** ;

Nome: **string**[25] ;

Risultato: **boolean** ; (definisce un dato che può essere TRUE o FALSE)

Istruzioni di ingresso e uscita

- Con l'istruzione Write (con gli apici) viene mandato sul video il messaggio indicato tra apici.

Write ('messaggio');

- L'istruzione Write (senza apici) visualizza il valore assunto dalla variabile, lasciando il cursore sulla stessa riga; l'output di una successiva istruzione verrà visualizzato subito dopo.

Write (variabile);

L'istruzione Writeln porta il cursore all'inizio della riga successiva del video (cioè va a capo); l'output successivo comparirà all'inizio di una nuova riga.

- Se si vogliono ottenere i risultati tramite stampante, invece di visualizzarli su video, occorre aggiungere la dichiarazione

Uses Printer ;

e, davanti al messaggio o alla variabile da stampare, occorre aggiungere la parola **Ist**

Write (Ist, variabile);

- In TP i numeri reali vengono scritti su schermo o su stampante in forma esponenziale; se si vuole che i numeri reali in uscita vengano scritti in forma diversa, occorre dichiarare, dopo il nome della variabile, quanti caratteri si vogliono vedere scritti e quante cifre decimali si vogliono dopo la virgola.

Es.: Writeln (A:8:2);

- L'istruzione Readln acquisisce dalla tastiera un valore e lo assegna alla variabile avente il nome specificato.

Readln (nomevariabile);

Se si vogliono leggere più dati, si devono indicare nelle parentesi i nomi di tante variabili quanti sono i dati digitati prima di Invio.

```
IF condizione
  THEN
  Istruzione-a
  ELSE
  Istruzione-b;
```

Le istruzioni a e b verranno eseguite alternativamente, ossia se la condizione è vera viene eseguita l'istruzione a e non la b, mentre se la condizione è falsa, viene eseguita l'istruzione b e non la a.

L'istruzione

IF...THEN...ELSE...;

Costituisce una istruzione unica, pertanto prima e dopo ELSE non si deve mettere il punto e virgola.

FOR K := valore iniziale TO valore finale DO

L'elaboratore ripete le istruzioni comprese fra il begin e l'end (ciclo) per un determinato numero di volte; la variabile contatore K si incrementa di uno ogni volta che il ciclo viene eseguito. Quando la variabile contatore supera il valore finale, l'elaboratore termina l'esecuzione del ciclo ed esegue le istruzioni successive. ES: PROGRAM Media;

REPEAT ... UNTIL (condizione)

L'elaboratore ripete tutte le istruzioni comprese tra le due, finché non è verificata la condizione. Quando la condizione è verificata, esegue le istruzioni successive. ES.: PROGRAM TOTALE;

WHILE (condizione) DO

L'elaboratore esegue le istruzioni comprese fra il begin e l'end fino a quando la condizione è verificata.
ES.: PROGRAM EUCLIDE;

L'istruzione:

End.

fa terminare l'esecuzione del programma da parte dell'elaboratore.

```
PROGRAM Media_di_n_numeri;
USES WINCRT;
VAR Somma, X, Media : real;
    K, n : integer;
BEGIN
    WRITELN ( ' Inserisci il valore di n');
    READLN (n);
    Somma := 0;
    FOR K:=1 TO n DO
        BEGIN
            WRITELN ( ' Scrivi il ', k, ' numero' );
            READLN ( X);
            Somma:= Somma + X;
        END;
    Media:= Somma / n;
    WRITELN ( ' Media = ', Media :8:2);
END.
```

```
PROGRAM Totale_spesa;  
Uses Wincrt;  
VAR Totale, Prezzo : integer;  
BEGIN  
    WRITELN ('Scrivi i prezzi dei prodotti e, per terminare, scrivi 0');  
    WRITELN;  
    Totale :=0;  
    REPEAT  
        WRITELN ('Scrivi il prezzo');  
        READLN ( Prezzo );  
        TOTALE := Totale+ Prezzo;  
    UNTIL Prezzo=0;  
    WRITELN (' Totale spesa= ', Totale);  
END.
```

```
PROGRAM EUCLIDE;  
USES WINCRT;  
VAR A, B, MCD: integer;  
BEGIN  
  WRITELN ( ' Inserisci il primo numero ' );  
  READLN (A);  
  WRITELN ( ' Inserisci il secondo numero ' );  
  READLN (B);  
  WHILE A<>B DO  
    BEGIN  
      IF A>B THEN A:=A-B  
        ELSE B:=B-A;  
    END;  
  MCD := A;  
  WRITELN ( 'MCD = ', MCD );  
END.
```

La programmazione strutturata

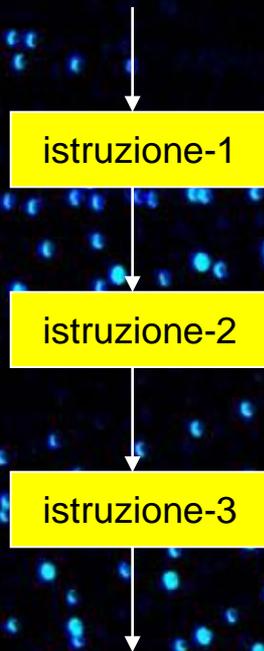
La programmazione strutturata è la progettazione e la realizzazione di un programma costituito da parti che dipendono l'una dall'altra secondo un ben definito modello organizzativo.

Qualsiasi algoritmo appropriato può essere scritto utilizzando soltanto tre strutture di base: **sequenza, alternativa, ripetizione.**

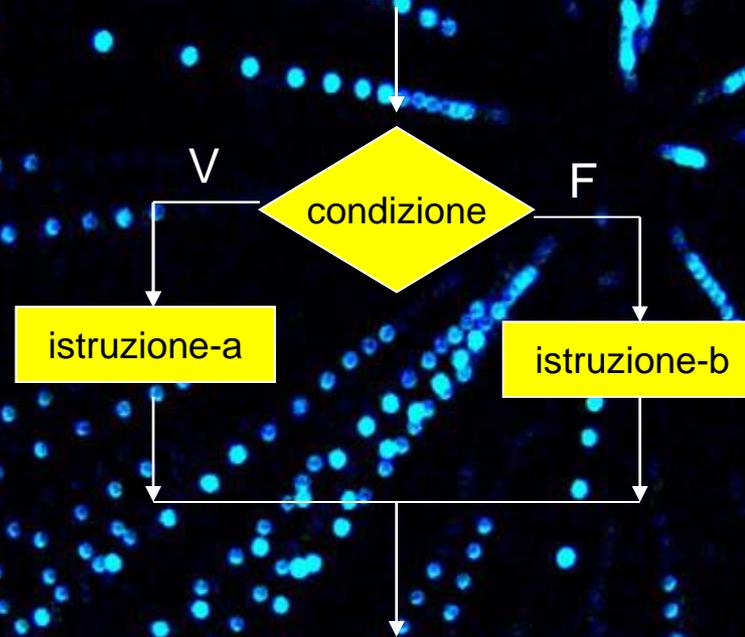
Questi tre modelli organizzativi di base si chiamano **strutture di controllo.**

Strutture di controllo

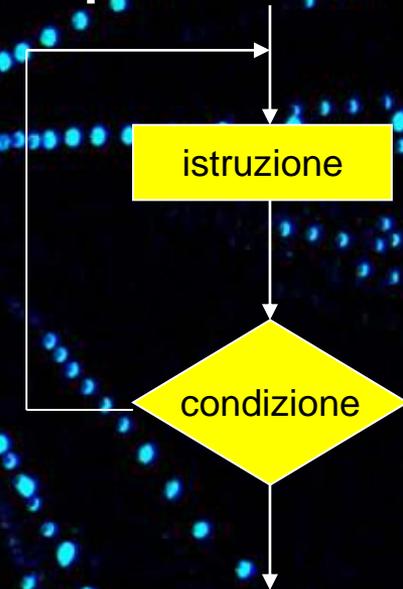
Sequenza



Alternativa



Ripetizione



Teorema di Böhm-Jacopini

Ogni algoritmo, scritto usando le istruzioni di salto, è anche rappresentabile usando soltanto le tre strutture di sequenza, selezione e ripetizione.

La sequenza in Pascal

BEGIN

istruzione-a;

istruzione-b;

Istruzione-c;

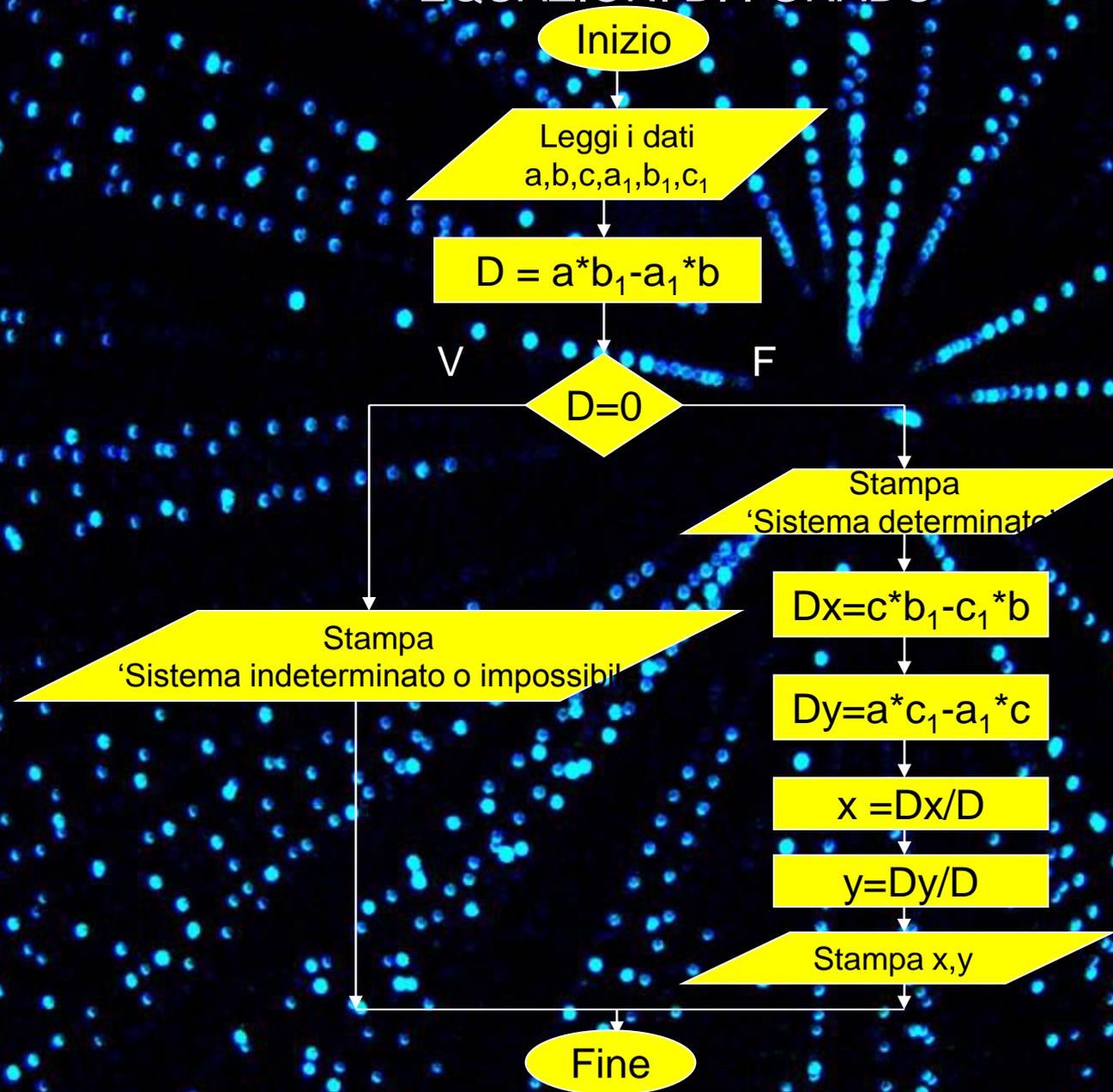
END.

Le istruzioni comprese nella sequenza verranno eseguite certamente una dopo l'altra.

PROGRAMMA IN PASCAL PER LA RISOLUZIONE DI UNA EQUAZIONE DI II GRADO

```
PROGRAM Equazione_di_secondo_grado;
USES WINCRT;
VAR a, b, c, Delta, x1, x2: real ;
BEGIN
  WRITELN ('Inserire i coefficienti dell'equazione a,b,c');
  READLN (a, b, c);
  IF a<>0 THEN
    BEGIN
      Delta:= sqr(b)-4*a*c;
      IF Delta<0 THEN
        WRITELN ( 'L'equazione non ammette soluzioni reali')
      ELSE
        BEGIN
          x1:=(-b-sqrt(Delta))/(2*a);
          x2:=(-b+sqrt(Delta))/(2*a);
          WRITELN ('x1 = ', x1);
          WRITELN ('x2 = ', x2);
        END;
      END;
    ELSE
      WRITELN ('Equazione di I° grado');
  END.
```

ALGORITMO PER LA RISOLUZIONE DI UN SISTEMA DI EQUAZIONI DI I GRADO



PROGRAMMA IN PASCAL PER LA RISOLUZIONE DI UN SISTEMA DI EQUAZIONI DI I GRADO

```
PROGRAM Sistema_di_equazioni_di_primo_grado ;
USES WINCRT;
VAR a, b, c, a1, b1, c1, D, Dx, Dy, x, y: real;
BEGIN
  WRITELN ('Inserire i coefficienti delle equazione a,b,c,a1,b1,c1');
  READLN (a, b, c, a1, b1, c1) ;
  D:= a*b1-a1*b ;
  IF D=0 THEN
    WRITELN ('Sistema indeterminato o impossibile')
  ELSE
    BEGIN
      Dx:=c*b1-c1*b;
      Dy:=a*c1-a1*c;
      x:=Dx/D;
      y:=Dy/D;
      WRITELN('x=',x:0:2);
      WRITELN('y=',y:0:2);
    END;
  END.
```

USO DELLE PROCEDURE

```
PROGRAM Equazione_di_secondo_grado;
USES WINCRT;
VAR a, b, c: integer;
    Delta, x1, x2: real ;

PROCEDURE Equazione_di_primo_grado;
BEGIN
  WRITELN('Equazione di primo grado');
  IF b=0 THEN
    IF c=0 THEN
      WRITELN('Equazione indeterminata')
    ELSE
      WRITELN('Equazione impossibile')
    ELSE
      BEGIN
        x1 := -c/b;
        WRITELN ('x = ', x1);
      END;
    END;
END;
```

```
BEGIN
  WRITELN ('Inserire i coefficienti dell"equazione a,b,c');
  READLN (a, b, c);
  IF a<>0 THEN
    BEGIN
      Delta:= sqr(b)-4*a*c;
      IF Delta<0 THEN
        WRITELN ( 'L"equazione non ammette soluzioni reali')
      ELSE
        BEGIN
          x1:=(-b-sqrt(Delta))/(2*a);
          x2:=(-b+sqrt(Delta))/(2*a);
          WRITELN ('x1 = ', x1);
          WRITELN ('x2 = ', x2);
        END;
      END;
    ELSE
      Equazione_di_primo_grado;
    END.
END.
```