Come ragiona il computer

Problemi e algoritmi

Il problema

- Abbiamo un problema quando ci poniamo un obiettivo da raggiungere e per raggiungerlo dobbiamo mettere a punto una strategia
- Per risolvere il problema individuiamo una sequenza di istruzioni elementari che, partendo dai dati noti, arrivi a dare la soluzione


Dati in ingresso
Elaborazione
Dati in uscita

I problemi tipici dell'informatica

Ricerca di informazione

- Trovare il numero di telefono di una persona, individuare il numero più piccolo di una sequenza, stabile se una parola precede alfabeticamente un’altra

Problemi di elaborazione di informazioni

- Calcolare il costo totale di un certo numero di prodotti, trovare perimetro e area di una figura geometria, …

Problemi di decisione

- Decidere se per andare a scuola è più conveniente il motorino, l'autobus, andare a piedi, farsi accompagnare da un genitore

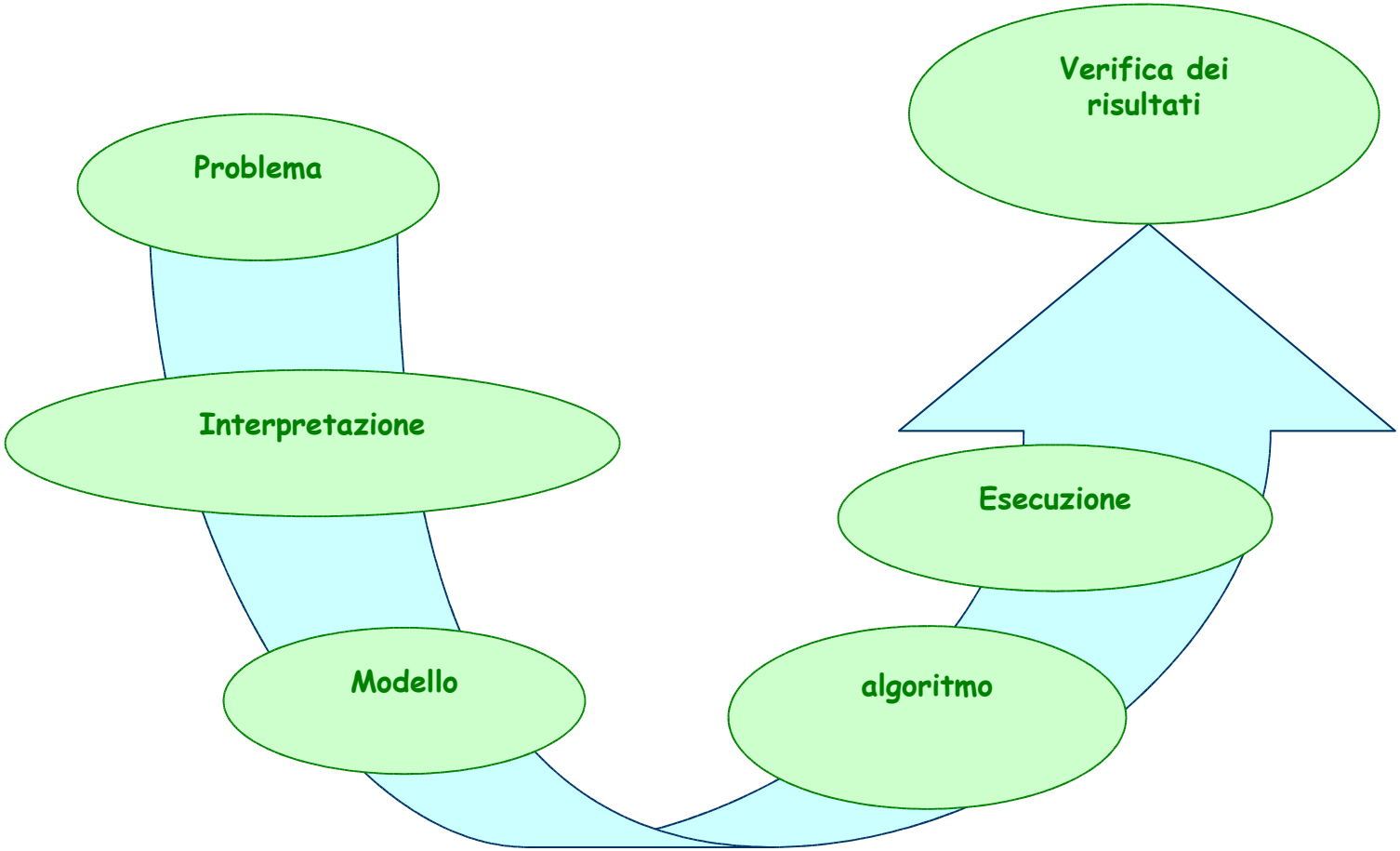
Problemi di ottimizzazione

- Trovare tra tutte le soluzioni possibili del problema quella che rende minimo un certo fattore, per esempio scegliere il mezzo di trasporto più economico per andare a Parigi oppure quello con il quale si impiega meno tempo

Risolvere un problema

- Interpretare l'enunciato
- Individuare i dati noti e quelli da trovare
- Costruire un modello
- Descrivere il procedimento risolutivo
- Eseguire le operazioni stabilite nel processo risolutivo
- Verificare se i risultati ottenuti corrispondono alla soluzione del problema reale

Risolvere un problema



Algoritmo

- Un algoritmo è una *sequenza finita di operazioni elementari* che porta alla soluzione del problema.

Diagramma di flusso o diagrammi a blocchi

- È uno metodi più comuni usati per la rappresentazione di algoritmi.
- Si presenta come un insieme di figure geometriche collegate da frecce.

Inizio

- Tutti i diagrammi a blocchi cominciano con un’ellisse che contiene la parola inizio

Dati in ingresso

- I dati in ingresso sono i dati noti del problema, quelli che devono essere elaborati per arrivare alla soluzione

Operazioni

- Le operazioni da svolgere sui dati sono racchiuse in rettangoli

Scelta

- Quando si deve fare una scelta tra due possibilità si usa il rombo

Dati in uscita

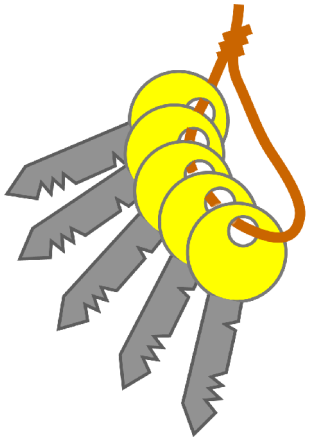
- I dati in uscita sono quelli che si vuole conoscere e costituiscono il risultato dell'elaborazione

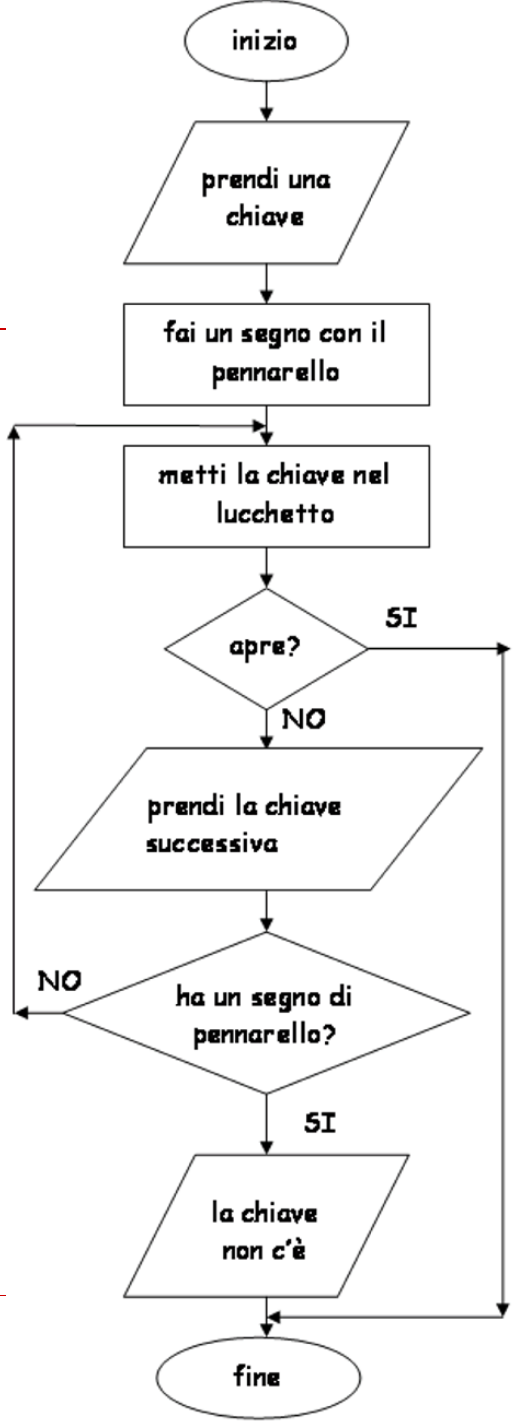
Fine

- Ogni diagramma di flusso si conclude con un’ellisse che contiene la parola fine

Problema della chiave

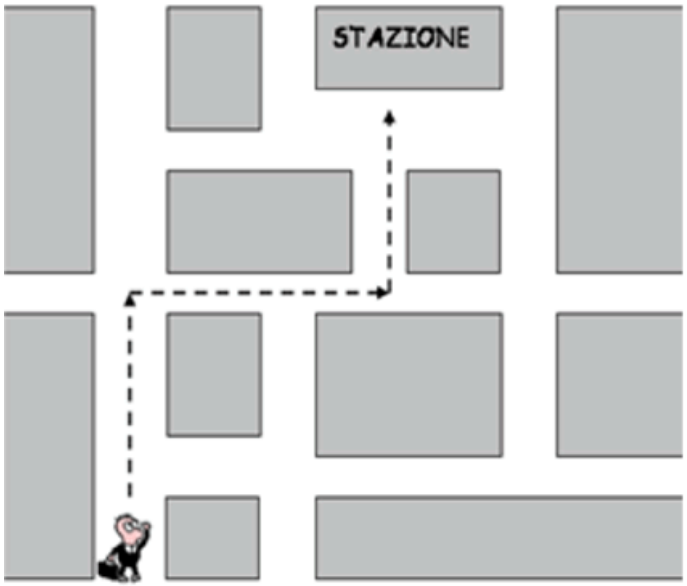
- Trovare in un mazzo di chiavi quella che apre il lucchetto

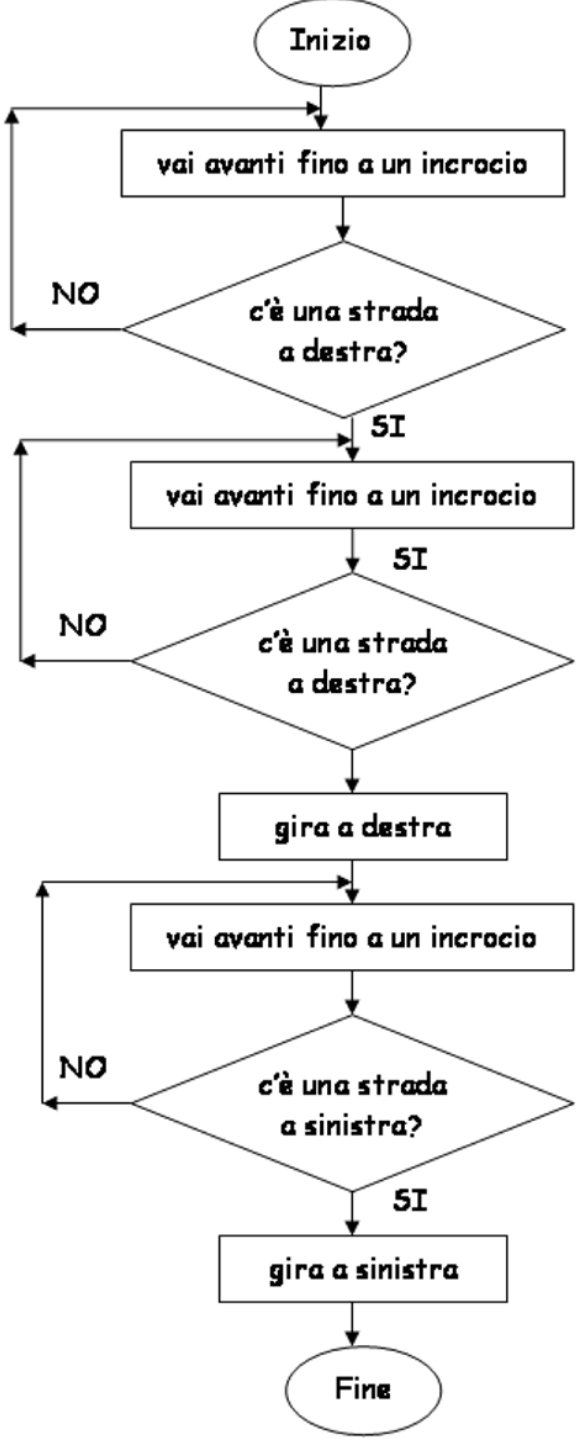




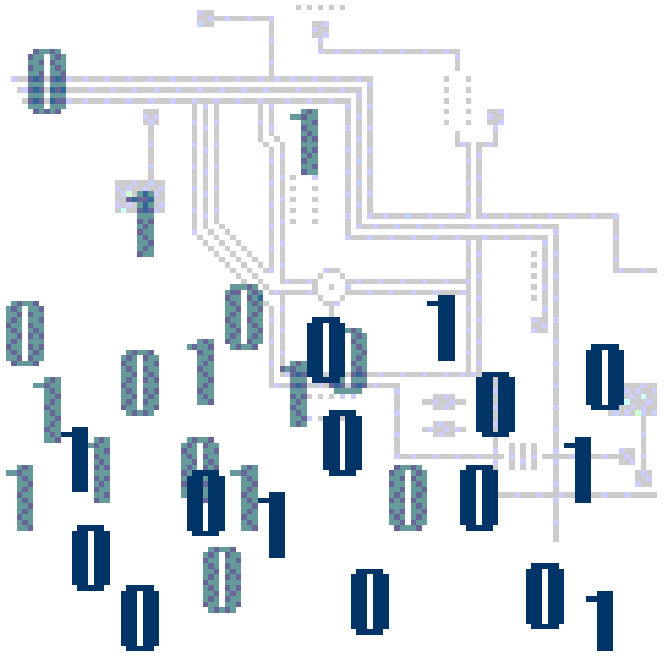
Problema della stazione

- Qual è la strada più breve per la stazione?





Programmare il computer



Dal problema all'algoritmo

- Per tradurre un problema in un algoritmo risolutivo occorre non solo saper risolvere il problema ma anche saperne descrivere il procedimento risolutivo.
- Il primo passo è individuare i dati del problema e distinguerli in dati in ingresso e dati in uscita.

Il problema della somma

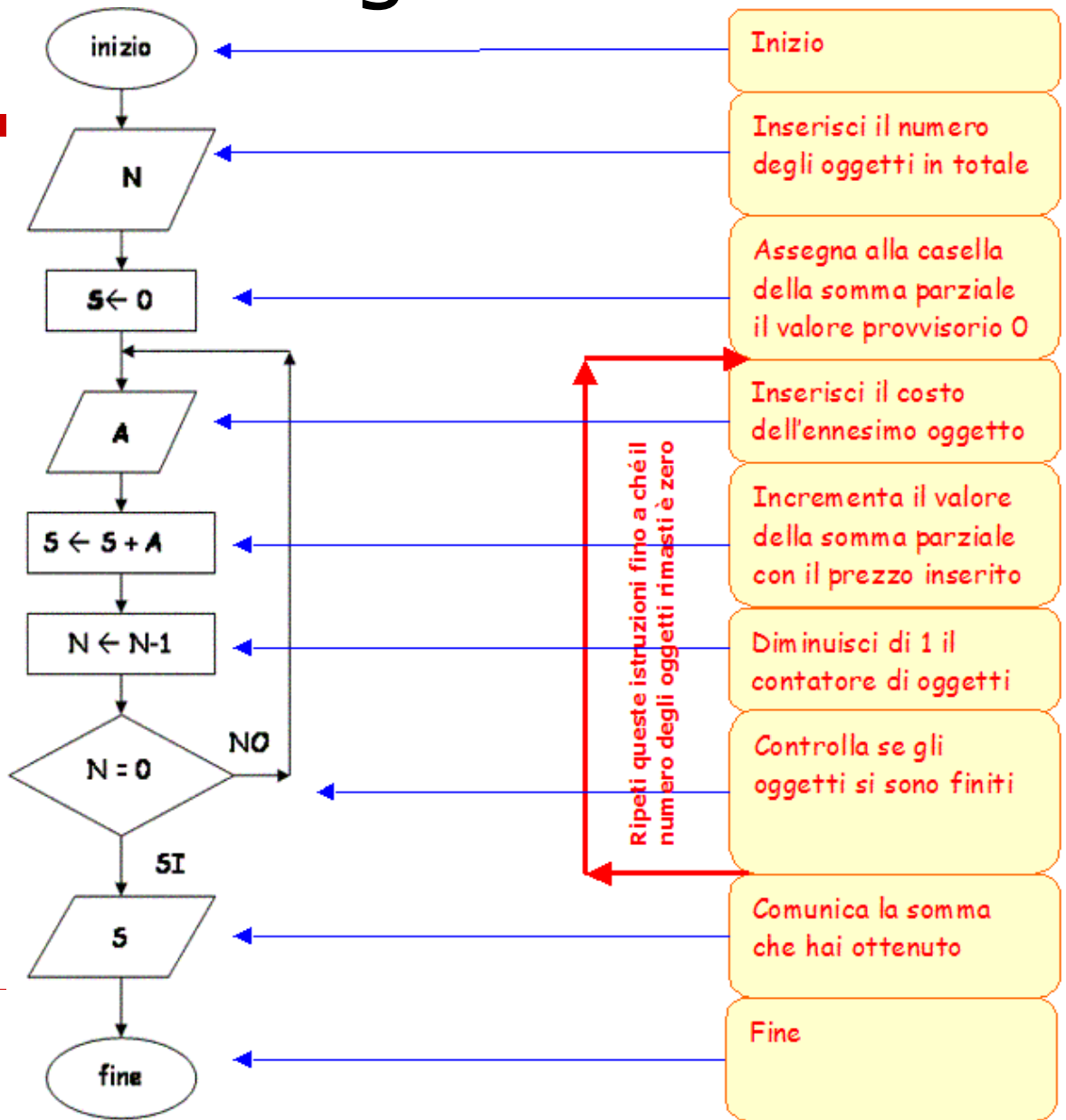
- Mario è stato in cartoleria e ha comprato i libri di matematica, di italiano, di scienze, un quaderno e una penna. Si pone il problema di sapere quanto ha speso in tutto.
- Si dicono **dati in ingresso** i dati noti che contribuiscono alla soluzione del problema. Si dicono **dati in uscita** quelli che si vogliono conoscere.

Come si risolve

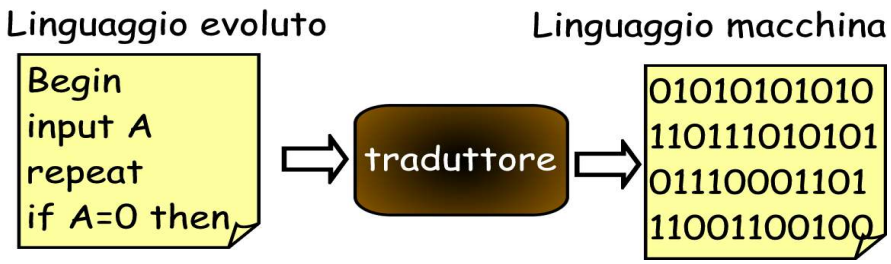
- **Problema generale**: acquistati un certo numero di oggetti, noti i singoli prezzi, calcolare il costo totale.
- Per risolvere questo problema generale occorre scrivere un **programma** che permetterà al computer di risolvere ogni situazione specifica.
- Per arrivare a scrivere il programma devi saper descrivere il procedimento risolutivo per mezzo di un **algoritmo** e tradurre l'algoritmo in un **linguaggio di programmazione** comprensibile al computer

Costruzione dell'algoritmo

Uno dei modi più comuni, usati per la rappresentazione di algoritmi è il **diagramma a blocchi**, o **diagramma di flusso**, in inglese flow-chart.



Linguaggio macchina e compilatori



Con la parola **programma** si intende una sequenza finita di istruzioni che il computer deve eseguire, una alla volta e nell'ordine prestabilito.
Tutti i programmi, di gioco, di scrittura, di calcolo, di grafica per essere comprensibili al computer devono essere codificati in una sequenza di simboli 0 e 1. Un linguaggio di questo tipo è detto **linguaggio macchina**, l'unico che la CPU è in grado di comprendere. All'inizio della storia dell'informatica, intorno alla metà del secolo scorso, gli specialisti dialogavano con i computer solo per mezzo del linguaggio macchina. Con il passare degli anni sono stati creati dei linguaggi più vicini al modo di comunicare dell'uomo. La traduzione di questi linguaggi più evoluti nel linguaggio macchina viene effettuato da programmi detti **traduttori** o **compilatori**.

Struttura di un programma in Pascal

- Un programma Pascal deve essere strutturato in tre sezioni: intestazione,dichiarazioni, istruzioni.
- Nell'**intestazione** si dà il nome al programma: si scrive la parola **PROGRAM** seguita dal nome del programma e dal punto e virgola. Il punto e virgola va inserito alla fine di ogni istruzione del programma, ad eccezione di pochi casi.
- Nella sezione delle **dichiarazioni** si definiscono tutti i dati che verranno utilizzati nel programma, suddivisi per costanti, variabili e altre categorie.
- Il gruppo delle **istruzioni** comincia con la parola riservata **BEGIN** e termina con la parola **END** seguita da un punto

Istruzioni fondamentali

- Le istruzioni di output o uscita sono **WRITE** (scrivi) e **WRITELN** (scrivi e vai a capo), hanno come effetto quello di far apparire sullo schermo quanto è scritto nelle parentesi se il testo è racchiuso tra apici, oppure il valore della variabile indicata nelle parentesi.
- Le istruzioni di input o immissione dati sono **READ** (leggi) e **READLN** (leggi e vai a capo).
- L'assegnazione delle variabili si ottiene con i segni **:=**, per cui C←5 si scrive in Pascal C:=5.

Un semplice programma…

Noto il raggio calcolare l'area del cerchio e la lunghezza della circonferenza.

```
PROGRAM cerchio;
CONST
pigreco = 3.14;
VAR
R, C, A : real;
WRITELN('Questo programma permette di calcolare circonferenza e area di un cerchio di raggio dato');
WRITE ('Inserisci il valore del raggio');
READLN(R);
C:=2*pigreco*R;
A:=pigreco*R*R;
WRITELN('Circonferenza =',C,'Area=',A);
READLN;
END.
```

...cosa significa

WRITELN('Questo programma permette …') farà apparire sul monitor esattamente il testo scritto tra parentesi. Lo scopo è quello di comunicare all'operatore che cosa fa il programma in esecuzione sul computer.

WRITE ('Inserisci il valore del raggio') ha lo scopo di comunicare all'operatore che deve inserire il valore numerico del raggio.

READLN(R) ha lo scopo di far leggere al computer e quindi assegnare alla variabile R il numero che digiteremo da tastiera.

C:=2*pigreco*R ha lo scopo di assegnare alla variabile C il valore ottenuto moltiplicando 2 per il valore della costante pigreco per il valore della variabile R (il simbolo per la moltiplicazione è *).

A:=pigreco*R*R ha lo scopo di assegnare alla variale A il valore ottenuto moltiplicando il valore di pigreco per il valore di R al quadrato

WRITELN('Circonferenza =',C,'Area=',A) ha lo scopo di far apparire sullo schermo il testo Circonferenza = seguito dal valore della variabile C, poi il testo Area= seguito dal valore della variabile A.

READLN è un'istruzione di attesa che permette di vedere i risultati dell'elaborazione. Se non ci fosse questa istruzione la visualizzazione sarebbe così rapida che non vedremmo niente.