Prerequisiti:
- Conoscenza degli insiemi dei numeri naturali e interi.
- Acquisizione del concetto di successione.

Questa unità è riservata agli Istituti Tecnici e agli Istituti Professionali (5ª classe).

OBIETTIVI DI APPRENDIMENTO

Una volta completata l'unità, gli allievi devono essere in grado di:

- *fornire una definizione non formalizzata di algoritmo*
- *definire e applicare in casi semplici le strutture" sequenza", "scelta", "ciclo"*
- *fornire una definizione del ciclo "for"*
- *spiegare cos'è un algoritmo iterativo e cosa un algoritmo ricorsivo*
- *elaborare un algoritmo risolutivo di qualche semplice problema*

85.1 Algoritmo.
85.2 Notazione lineare strutturata.
85.3 Algoritmo iterativo.
85.4 Algoritmo ricorsivo.
85.5 Particolari successioni ricorsive.
Verifiche.
Una breve sintesi per domande e risposte.

Algoritmi
Unità 85

85.1 ALGORITMO
85.2 NOTAZIONE LINEARE STRUTTURATA

Per questi due paragrafi si rimanda ai paragrafi 28.1 e 28.2 dell'unità 28, indirizzata ai soli Licei.

85.3 ALGORITMO ITERATIVO

85.3.1 La struttura "ciclo", sia a controllo in testa sia a controllo in coda, è quella di un algoritmo formato da una sequenza di istruzioni che sono ripetute ciclicamente sotto una determinata condizione.
Ogni algoritmo di questo tipo si chiama **algoritmo iterativo**.
Vogliamo fornire qui alcuni esempi più interessanti e significativi di quelli che abbiamo proposto in fase di presentazione del ciclo. Si tratta ovviamente di esempi di algoritmi che risolvono problemi. Utilizzeremo soltanto il "ciclo mentre", vale a dire il ciclo a controllo in testa.

85.3.2 ESEMPIO 1. **Calcolare la somma dei primi n numeri naturali a partire da 1**.
Si capisce che questa somma è:

$$s = 1 + 2 + 3 + \cdots + n.$$

Quando si esegue l'operazione con un automa, il valore di n non può però restare indeterminato ma bisogna precisarlo: supponiamo di fissare n=10.
Ebbene la forma retorica dell'algoritmo iterativo che risolve il problema è la seguente:

```
INIZIO
  ad n assegna il valore 0
  ad s assegna il valore 0
  MENTRE n è minore o uguale a 10 ESEGUI
    INIZIO
      ad n assegna il valore n+1
      ad s assegna il valore s+n
    FINE
    scrivi il valore di s
FINE
```

mentre lo schema di calcolo, scritto in NLS, è questo:

```
INIZIO
  n := 0
  s := 0
  MENTRE n≤10 ESEGUI
    INIZIO
      n := n+1
      s := s+n
    FINE
    scrivi(s)
FINE
```

Per verificare che l'algoritmo è effettivamente quello che risolve il nostro problema possiamo compilare quella che si chiama comunemente una ***tabella di traccia***, vale a dire una tabella nella quale si assegnano ad n ed s i valori che si trovano via via applicando la procedura insita nell'algoritmo (Tab. 1); il valore di s è anche quello che si ha all'uscita di ogni ciclo, vale a dire la somma dei primi n numeri, dove n è il valore segnalato nella colonna corrispondente della tabella.

n	1	2	3	4	5	6	7	8	9	10
s	1	3	6	10	15	21	28	36	45	55

TAB. 1

85.3.3 ESEMPIO 2. **Calcolare il fattoriale di un numero n**.

Ricordiamo che il fattoriale di un numero naturale n, indicato com'è noto con n!, si definisce nel modo seguente:

$$n! = \begin{cases} 1 \text{ se } n=0 \text{ oppure } n=1 \\ \underbrace{1 \cdot 2 \cdot 3 \cdot \ldots \cdot n}_{\text{n fattori}} \text{ se } n>1 \end{cases}$$

Ancora una volta bisogna precisare il valore di n. Supponiamo di fissare n=10.

Lo schema dell'algoritmo è molto simile al precedente: lì bisognava calcolare la somma dei primi 10 numeri a partire da 1; qui bisogna calcolare il prodotto dei primi 10 numeri a partire da 1.

Lasciamo a te sia la stesura della forma retorica dell'algoritmo sia quella in NLS. Compilerai pure una tabella di traccia per verificare se hai operato correttamente.

85.3.4 Un particolare tipo di algoritmo iterativo è il cosiddetto "ciclo FOR". Impone l'esecuzione di una sequenza di istruzioni ripetuta per un determinato numero n di volte, senza sottostare ad alcuna condizione a differenza, ad esempio, del "ciclo mentre". È basato essenzialmente su una variabile intera i, detta *contatore*, che viene incrementata di un'unità in ogni ciclo. La sua struttura, posto che la variabile contatore parta da 1, è la seguente:

```
PER i che varia da 1 ad n
ESEGUI istruzioni
```

- Un esempio banale è costituito dal seguente algoritmo che fornisce in uscita una successione di 10 simboli uguali ad 1:

```
PER i che varia da 1 a 10
ESEGUI
  scrivi il simbolo 1
```

- Un esempio un po' più interessante è correlato al problema di:

  «scrivere i primi n termini di una progressione aritmetica di primo termine a e ragione d».

La progressione è evidentemente la seguente:

$$\mathbf{a},\ \ \mathbf{a+d},\ \ \mathbf{a+2d},\ \ \mathbf{a+3d},\ \ \ldots,\ \ \mathbf{a+(n-1)d},\ \ \ldots\ .$$

Un algoritmo idoneo a determinarla, ovviamente dopo aver assegnato i valori di n, a, d, è il seguente, dove si è posto n=10, a=2, d=3:

```
INIZIO
  A0 := 2
  D := 3
  PER i che varia da 1 a 10 ESEGUI
    INIZIO
```

```
      AN := A0+(i–1)D
      scrivi il valore di A
    FINE
FINE
```

Lasciamo a te la compilazione di una tabella di traccia.

- Un esempio ancora più interessante e significativo di ciclo FOR è correlato alla celebre successione di Fibonacci [1]:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, … , $f_{n-2}+f_{n-1}$, … .

La regola per ottenerla è semplice: i primi due numeri sono entrambi uguali ad 1 mentre ogni altro numero di posto n è la somma dei due immediatamente precedenti, cioè di quelli di posto n–2 ed n–1.
Un algoritmo idoneo a determinarla, ovviamente una volta fissato n, è il seguente, dove si è stabilito per n il valore 15:

```
INIZIO
  A := 1
  B := 0
  PER i che varia da 2 a 15 ESEGUI
    INIZIO
      Fib := A+B
      B := A
      A := Fib
      scrivi il valore di Fib
    FINE
FINE
```

Ecco qui appresso la tabella di traccia (Tab. 2).

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Fib	1	2	3	5	8	13	21	34	55	89	144	233	377	610
x2	1	1	2	3	5	8	13	21	34	55	89	144	233	377
x1	1	2	3	5	8	13	21	34	55	89	144	233	377	610

TAB. 2

85.4 ALGORITMO RICORSIVO

85.4.1 Un'altra tipologia di algoritmi è costituita dagli *algoritmi ricorsivi*.
Per comprendere di cosa si tratti mettiamo a confronto gli ultimi due esempi considerati sopra.
Nel primo di essi una formula risolve la questione: $a_n=a+(n-1)d$.
Nel secondo non c'è una formula, bensì un procedimento atto a determinare i vari termini. Procedimento che possiamo sintetizzare nel modo seguente:

[1] $$a_n = \begin{cases} 1 \text{ se } n=1 \text{ oppure } n=2 \\ a_{n-2} + a_{n-1} \text{ se } n>2 \end{cases}$$

Se ne desume che il calcolo del termine di posto n presuppone noti quelli di posto n–1 ed n–2, il cal-

[1] **Fibonacci**, Leonardo, detto anche Pisano, 1175-1235.

colo del termine di posto $n-1$ a sua volta presuppone noti quelli di posto $n-2$ ed $n-3$ e così via andando a ritroso fino al calcolo di a_3 che è uguale ad a_1+a_2 e perciò $a_3=1+1=2$. Da qui, poi, si risale verso a_n. Ragion per cui, con questa definizione, non c'è modo di calcolare direttamente a_n ma bisogna passare per i termini precedenti.
Una successione come quella di Fibonacci ha dunque la caratteristica di richiamare se stessa nella costruzione dei vari termini e per questo si definisce *successione ricorsiva* e l'algoritmo idoneo a costruirla si definisce a sua volta *algoritmo ricorsivo*.
In generale:

Un **algoritmo ricorsivo**
è un algoritmo che richiama se stesso nel calcolo dei vari termini.

Un algoritmo ricorsivo è costituito da due passi fondamentali:

- Il primo passo o *passo base* fissa il risultato corrispondente al valore iniziale (o eventualmente ai valori iniziali se sono più d'uno);
- Il secondo passo o *passo induttivo* stabilisce il risultato corrispondente all'indice n in funzione di quello corrispondente all'indice precedente $n-1$ (o eventualmente ad indici precedenti).

La sua struttura è sostanzialmente la struttura della "scelta binaria" (vedi 28.2.3), dove adesso la "condizione" è costituita dal valore iniziale, l'istruzione B_1 è il passo base e l'istruzione B_2 è il passo induttivo.

Forniamo, a titolo di esempio, la rappresentazione grafica del diagramma a blocchi dell'algoritmo che genera la successione di Fibonacci (Fig. 1).

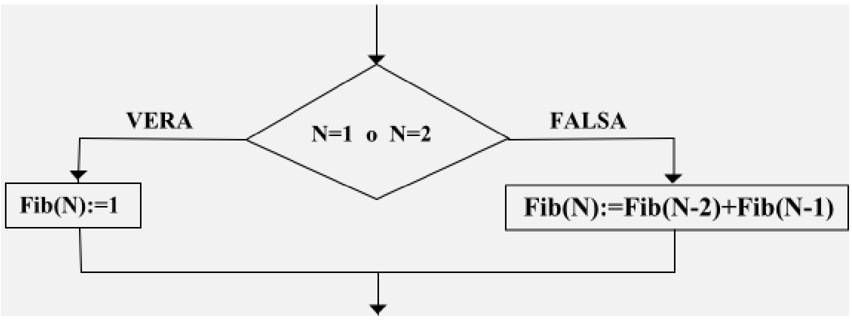

FIG. 1

La NLS del medesimo algoritmo è la seguente:

```
INIZIO
  ...
  SE N=1 o N=2 ALLORA Fib(N) := 1
  ALTRIMENTI Fib(N) := Fib(N–2)+Fib(N–1)
  ...
FINE
```

85.4.2 Si possono fornire numerosi esempi di algoritmi ricorsivi, ma ci limitiamo ad alcuni soltanto e precisamente a quelli che risolvono problemi che sono stati risolti sopra con algoritmi iterativi. Si tratta sostanzialmente di passare dalla forma analitica della formula che risolve il problema alla forma ricorsiva. Per la successione di Fibonacci non abbiamo avuto bisogno di operare questa trasformazione giacché la successione è già data in forma ricorsiva. Per gli altri algoritmi però la trasformazione è necessaria. Ce ne andiamo ad occupare.

- ESEMPIO 1. **Esprimere in forma ricorsiva la funzione che fornisce somma dei primi n numeri naturali a partire da 1**.

La somma s(n) dei primi n numeri naturali a partire da 1 è evidentemente una funzione di n. Precisamente:

$$s(n) = 1 + 2 + 3 + 4 + \ldots + n.$$

In realtà conosciamo la forma analitica della formula che fornisce la somma s(n). Esattamente:

$$s(n) = \frac{n(n+1)}{2}.$$

Ma non è questa formula che ci interessa, poiché chiaramente non è una forma ricorsiva.
Ora, una volta constatato che s(1)=1, è abbastanza semplice capire che risulta:

- s(2) = 1+2 = s(1)+2,
- s(3) = s(2)+ 3,
- s(4) = s(3)+4,

e così via.
In generale, se n>1, si ha s(n) = s(n–1)+n.
Per cui la forma ricorsiva della funzione s(n) è la seguente:

$$s(n)=\begin{cases}1 \text{ se } n=1\\ s(n-1)+n \text{ se } n>1\end{cases}$$

Prova a scrivere l'algoritmo che risolve il problema di determinare la somma dei primi 100 numeri naturali a partire da 1.

- ESEMPIO 2. **Esprimere in forma ricorsiva la funzione che fornisce il fattoriale di un numero n**.

Bisogna ricordare che si ha: $n!=n(n+1)$. Per cui la forma ricorsiva richiesta è la seguente:

$$n!=\begin{cases}1 \text{ se } n=0\\ (n-1)!n \text{ se } n>0\end{cases}$$

Prova a scrivere l'algoritmo che genera il fattoriale di 10.

- ESEMPIO 3. **Esprimere in forma ricorsiva la formula che fornisce l'n-esimo termine di una progressione aritmetica**.

L'n-esimo termine a_n di una progressione aritmetica di 1° termine a e ragione d è dato dalla seguente espressione analitica:

$$a_n=a+(n-1)d.$$

La definizione ricorsiva di a_n si ottiene constatando semplicemente che ogni termine della progressione si ottiene dal precedente sommandogli d. Pertanto:

$$a_n=\begin{cases}a \text{ se } n=1\\ a_{n-1}+d \text{ se } n>1\end{cases}$$

Altri esempi te li proponiamo per esercizio.

Scrivi la forma ricorsiva delle seguenti successioni:

a) Somma dei primi n termini di una progressione aritmetica di primo termine a e ragione d.
b) n-esimo termine di una progressione geometrica di primo termine a e ragione q.
c) Somma dei primi n termini di una progressione geometrica di primo termine a e ragione q.
d) Somma dei primi n numeri naturali dispari a partire da 1.

e) Somma dei quadrati dei primi n numeri naturali a partire da 1.
f) Somma dei cubi dei primi n numeri naturali a partire da 1.

85.5 PARTICOLARI SUCCESSIONI RICORSIVE

Per questo paragrafo si rimanda all'unità 60, paragrafo 60.3, indirizzato al solo Liceo Scientifico. Gli studenti dei Tecnici e dei Professionali possono fermare la loro attenzione solamente sui primi due esempi, descritti in 60.3.1 e 60.3.2.
Un terzo esempio di successione ricorsiva riguarda il ***metodo delle tangenti*** quale **procedimento di approssimazione degli zeri di una funzione**. Per la sua trattazione rinviamo all'unità 69, ultima parte del paragrafo 69.3.2.

VERIFICHE

1. Sulla base del seguente algoritmo completa la tabella sottostante:

```
inizio
    leggi(a,b)
    a := a·a
    b := b·b
    c := a+b
    scrivi(c)
fine
```

a	0	1	1	3
b	0	1	2	4
c				

2. Sulla base del seguente algoritmo completa la tabella sottostante:

```
inizio
    leggi(a,b)
    a := 2·a
    b := b·b
    c := a·b
    scrivi(c)
fine
```

a	0	2	3	2
b	1	2	2	3
c				

3. Sulla base del seguente algoritmo completa la tabella sottostante:

```
inizio
    leggi(n)
    se n>5 allora
        a := n-5
```

```
    altrimenti
        a := n·n
    scrivi(a)
fine
```

n	3	5	7	10
a				

4. Stabilire che differenza c'è tra i due seguenti algoritmi, dopo aver trovato i risultati in uscita:

```
inizio
    a := 3
    b := 4
    a := b
    b := a
    scrivi(a,b)
fine
```

```
inizio
    a := 3
    b := 4
    c := a
    a := b
    b := c
    scrivi(a,b)
fine
```

5. Considera il seguente algoritmo:

```
inizio
a := 0
mentre a < 100 esegui
    inizio
        scrivi(a)
        a := a+2
    fine
fine
```

Quale problema risolve questo algoritmo? Individua l'unica alternativa corretta tra le seguenti.

[A] Permette di riconoscere se un numero è minore di 100.
[B] Permette di stabilire se un numero minore di 100 è pari.
[C] Genera i numeri pari minori di 100.
[D] Calcola la somma dei numeri pari minori di 100.

6. Che cosa calcola il seguente algoritmo, tenendo presente che n è un naturale non nullo?

```
inizio
leggi(n)
a := 1
p := 1
mentre a<n esegui
    inizio
        a := a+1
        p := p·a
    fine
    scrivi(p)
fine
```

7. Che cosa calcola il seguente algoritmo, tenendo presente che a, b sono numeri interi?

```
inizio
    leggi(a,b)
    x := a
    y := b
    mentre x≠y esegui
        se x<y allora
            x := x+a
        altrimenti
            y := y+b
    scrivi(x)
fine
```

8. Scrivi in NLS gli algoritmi che risolvano i problemi seguenti:

a) Determinare se un dato numero naturale n è pari oppure dispari.
b) Dato un numero naturale n non nullo, determinare tutti i suoi divisori.
c) Di ogni numero naturale minore di 15 scrivere il successivo se il numero è pari e il suo quadrato se è dispari.
d) Dati due numeri naturali a e b diversi da 0, calcolare il loro MCD ed il loro mcm.
e) Calcolare la somma dei primi n numeri naturali, a partire da 1.
f) Calcolare il prodotto dei primi n numeri naturali, a partire da 1.
g) Scrivere la successione dei multipli di a non maggiori di b.
h) Dati i due numeri naturali n e k, scrivere i primi n multipli di k.

9. Si consideri la successione di termine generale a_n, la quale genera numeri interi. Scrivere la successione in forma ricorsiva, sapendo che:

a) $a_n = 3n+1$, b) $a_n = n(n+1)$, c) $a_n = 2 \cdot 3^n$.

10. Definire per ricorsione la successione di termine generale $a_n = r^n$, dove r è un qualsiasi numero reale non nullo (si tratta evidentemente della definizione ricorsiva della **potenza di un numero con esponente naturale**).

11. Si consideri la successione di termine generale:

$$a_n = \begin{cases} \dfrac{1}{2} & \text{se } n=1 \\ a_{n-1} + \dfrac{1}{n(n+1)} & \text{se } n>1 \end{cases}$$

a) Esprimere a_n in forma analitica.
b) Dimostrare che per ogni naturale n risulta $a_n \neq 1$.
c) Dopo aver dimostrato che la successione di termine generale:

$$b_n = \frac{1+a_n}{1-a_n}$$

è una progressione aritmetica, calcolare la somma dei suoi primi 100 termini.

[**R.** ...; c) $S_{100} = 10200$]

12. Si consideri la successione di termine generale:

$$a_n=\begin{cases}1 \text{ se } n=0\\ \dfrac{a_{n-1}}{a_{n-1}+1} \text{ se } n>0\end{cases}$$

a) Esprimere a_n in forma analitica.
b) Dimostrare che per ogni naturale n risulta $a_n \neq 1/n$.
c) Dopo aver dimostrato che la successione di termine generale:

$$b_n = \frac{2 + a_n}{1 - n\, a_n}$$

è una progressione aritmetica, calcolare la somma dei suoi primi n termini.

[**R.** … ; c) $S_n=n(n+2)$]

13. Si consideri la successione di termine generale:

$$a_n=\begin{cases}2 \text{ se } n=1\\ a_{n-1}+(n-1) \text{ se } n>1\end{cases}$$

a) Esprimere a_n in forma analitica.
b) Dimostrare, mediante il principio d'induzione, che la somma S_n dei primi n termini della successione è:

$$S_n = \frac{n(n^2 + 11)}{6}.$$

$\left[\textbf{R. } \text{a) } a_n=2+\dfrac{n(n-1)}{2};\ldots\right]$

UNA BREVE SINTESI PER DOMANDE E RISPOSTE

DOMANDE.

1. Come si può definire, ancorché in modo non formalmente rigoroso, un algoritmo?
2. Quali sono le strutture fondamentali nella Notazione Lineare Strutturata?
3. Se la condizione del costrutto "RIPETI B FINCHÉ condizione" è inizialmente vera, quante volte è eseguito il ciclo?
4. Scrivi in NLS un algoritmo idoneo a determinare tutti i divisori di 60.
5. Cos'è un algoritmo ricorsivo?
6. Scrivere la forma ricorsiva della successione il cui termine generale esprime la somma dei cubi dei primi n numeri naturali a partire da 1.
7. Calcolare a_{20} sapendo che $a_n=\begin{cases}2 \text{ se } n=1\\ a_{n-1}+(n-1) \text{ se } n>1\end{cases}$
8. Il termine generale a_n di una successione è tale che $a_n = \begin{cases}1 \text{ se } n=0\\ \sqrt{1 + a_{n-1}} \text{ se } n>0\end{cases}$

   Mediante uno strumento di calcolo automatico verificare che la successione converge al numero aureo.

RISPOSTE.

1. Un algoritmo si può definire come una sequenza ordinata e finita di istruzioni, che siano effettivamente eseguibili.
2. Sono la "sequenza", la "scelta (binaria)" ed il "ciclo".

3. Si tratta del ciclo a controllo in coda e siccome si ripete ogni volta che la condizione è falsa, mentre termina appena la condizione è vera, si capisce che se la condizione iniziale è vera, il procedimento ha termine senza che venga eseguito alcun ciclo.

4. Un algoritmo idoneo a determinare tutti i divisori di 60 è il seguente:

```
INIZIO
    N := 60
    RIPETI
        Divisore := 60 div N
        Resto := 60 mod N
        SE Resto=0 scrivi(Divisore)
        N := N-1
    FINCHÉ N<1
FINE
```

5. Si definisce algoritmo ricorsivo un algoritmo che richiama se stesso nella costruzione dei vari termini.

6. Constatato che si ha: $a_n = 1^3+2^3+3^3+...+n^3$, è facile comprendere che la forma ricorsiva della successione è $a_n=\begin{cases}1 \text{ se } n=1\\ a_{n-1}+n^3 \text{ se } n>1\end{cases}$

7. Si constata anzitutto che si ha:

$$a_1=2,\ \ a_2=2+3,\ \ a_3=(2+3)+4,\ \ \dots,\ \ a_{20}=(2+3+4+...+20)+21;$$

ne consegue che a_{20} è la somma di 20 numeri in progressione aritmetica di primo termine 2 ed ultimo termine 21. È pertanto:

$$a_{20}=20\cdot\frac{2+21}{2}=230.$$

8. Bisogna arrivare ad a_8 per scoprire che appunto $a_8\approx1,618$.