

Consorzio Nettuno - Corso di Matematica 1

Schede di lavoro guidato per le esercitazioni

A cura di Sebastiano Cappuccio

SCHEDA N. 1

ARGOMENTO: Congetture e confutazioni. Principio di induzione.

(LEZIONE n. 1)



ATTIVITA' N. 1:

Dopo aver mandato in esecuzione il programma *DERIVE*, selezionare **Author**, digitare **vector**($n^2+n+41,n,1,10$) <↵>, quindi selezionare **Simplify**.

La funzione **vector** genera un vettore ¹ le cui componenti sono i valori che l'espressione indicata, in questo caso $n^2 + n + 41$, assume quando la variabile n assume valori che vanno da un minimo di 1 ad un massimo di 10, a passi di una unità ². Equivale ad un ciclo enumerativo in un linguaggio di programmazione.

Selezionare **Factor**.

Si noti che questa operazione non sembra produrre alcun risultato, se non quello di riscrivere il vettore: in effetti la funzione **Factor** realizza la fattorizzazione dell'espressione evidenziata; in questo caso tutte le componenti del vettore sono numeri primi. Questo ci conduce a formulare la congettura che il polinomio dato fornisca sempre un numero primo, $\forall n \in N$.

Selezionare **Author**; evidenziare con il tasto cursore <freccia su> la funzione digitata all'inizio di questa Attività. Premere il tasto funzione <F3>.

Come si è già visto, questa azione porta l'espressione evidenziata nella linea di editing. Modificare la funzione in modo che la variabile n assuma valori variabili da 11 a 20, premere <↵> per confermare, quindi ripetere i comandi **Simplify** e **Factor**.

Ancora una volta si ottengono numeri tutti primi; ciò non fa che confermare la validità della congettura formulata.

Ripetere le azioni precedenti (modifica dell'espressione, poi **Simplify** e **Factor**) in modo da far variare n da 21 a 30, poi da 31 a 40.

Si noti che l'ultima componente, corrispondente ad $n = 40$, si fattorizza in 41^2 .

Abbiamo trovato una confutazione della congettura formulata, che quindi si è rivelata, contrariamente alle nostre aspettative, falsa.

Trovare almeno altri due valori di n che forniscano valori non primi del polinomio

E' interessante osservare che il primo numero non primo ottenuto è proprio il quadrato del termine noto del polinomio. Si tratta forse di un risultato generale? Trovare almeno un esempio per confutare questa nuova congettura.



ATTIVITA' N. 2:

Si vuole determinare la somma dei primi 48 numeri naturali (escluso lo 0).

L'operazione $1 + 2 + 3 + \dots + 47 + 48$ può essere indicata in forma abbreviata con il seguente simbolo: $\sum_{k=1}^{48} k$, che si legge *sommatoria di k per k che va da 1 a 48*.

¹ Cioè una n -upla ordinata di valori, che *DERIVE* racchiude tra parentesi quadre.

² Si è scelto di far variare n a gruppi di dieci solo per ottenere numeri sulla stessa riga dello schermo.

Per calcolarla con *DERIVE*, selezionare **Author**, digitare **k** <↵>; selezionare **Calculus Sum**, premere <↵> per confermare di voler calcolare la sommatoria dell'espressione prima digitata, premere ancora <↵> per confermare che la variabile considerata è k . Premere ancora <↵> per confermare i valori proposti nel menu (**Lower limit: 1, Upper limit: n**).

Nella finestra di Algebra compare il simbolo di sommatoria che rappresenta la somma dei primi n numeri naturali.

Selezionare **Simplify**.

Si ottiene il valore desiderato.

Selezionare **Author**; digitare la seguente espressione: $f(n) := \frac{n(n+1)}{2}$.

Viene così definita la funzione $f(n)$ che rappresenta una nota formula; questa definizione resterà attiva durante tutta la sessione di lavoro.

Selezionare **Author** e digitare **n:=48** <↵>. Evidenziare con il tasto cursore <freccia su> la funzione prima definita ed infine selezionare **Simplify**.

Il comando **n:=48** costituisce una vera e propria assegnazione, attraverso la quale la variabile n assume il valore 48; tale valore resterà durante tutta la sessione di lavoro o fino ad una successiva assegnazione.

Si noti che la formula ora digitata fornisce proprio la somma dei primi 48 numeri naturali (escluso lo 0) calcolata in precedenza.

Ripetere la verifica per $n = 100$ e per $n = 258$.

Al termine digitare **n:=n** <↵>.

Questa assegnazione serve a "liberare" la variabile n , cioè a far sì che d'ora in poi la variabile torni ad essere manipolata simbolicamente e non numericamente con l'ultimo valore che le è stato assegnato. Se non eseguiamo questa assegnazione, *DERIVE* continuerebbe a "ricordare" l'ultimo valore assegnato ad n fino alla fine della sessione di lavoro.

In una prossima scheda vedremo un altro modo per effettuare una "assegnazione locale" ad una variabile, cioè una assegnazione che ha effetto per la sola espressione a cui si fa riferimento e che quindi non richiede ulteriori assegnazioni.



ATTIVITA' N. 3:

Si vuole verificare con *DERIVE* la verità della seguente congettura: *la somma dei primi n numeri dispari è uguale ad n^2 .*

Selezionare **Author**. Digitare **2k-1** <↵>. Selezionare **Calculus Sum**, premere <↵> per confermare di voler calcolare la sommatoria dell'espressione prima digitata, premere ancora <↵> per confermare che la variabile considerata è k . Premere ancora <↵> per confermare i valori proposti nel menu (**Lower limit: 1, Upper limit: n**).

Nella finestra di Algebra compare il simbolo di sommatoria $\sum_{k=1}^n (2k-1)$, che rappresenta la somma dei primi n numeri dispari.

Si noti infatti che l'espressione $2k-1$ è certamente un numero dispari per ogni k , anzi è proprio il k -esimo dei numeri dispari.

Selezionare **Author**, digitare **vector**(, quindi evidenziare con il tasto <freccia su> la sommatoria prima ottenuta, premere il tasto funzione <F3> per inserirla nella linea di editing, poi proseguire la digitazione con **,n,1,10**) <↵>.

Selezionare **Simplify**.

Questo comando fa comparire nella finestra di Algebra sotto forma di componenti di un vettore (V. Attività n. 1) la somma dei primi 1, 2, 3, ..., 9, 10 numeri dispari.

Si noti che sono tutti "quadrati perfetti", rispettivamente di 1, 2, 3, ..., 9, 10.

La congettura sembra essere vera.

Ripetere la verifica della congettura, ad esempio, per n che va da 11 a 20.



ATTIVITA' N. 4:

Visto che non è stata trovata una confutazione della congettura formulata nella precedente Attività, che d'ora in poi indicheremo con il simbolo $P(n)$, si dimostri per induzione la sua validità.

Come in ogni dimostrazione per induzione, iniziamo mostrando la cosiddetta *base dell'induzione*, cioè la validità della proposizione nel caso iniziale, $n=1$.

La proposizione $P(1)$ è certo vera: la somma che ha come (unico) addendo il primo dei numeri dispari, 1, è certo un quadrato perfetto.

Supponendo vera la proposizione $P(n)$, cioè che

$$\sum_{k=1}^n (2k-1) = n^2,$$

dimostriamo che è vera la proposizione $P(n+1)$, cioè la proposizione:

$$\sum_{k=1}^{n+1} (2k-1) = (n+1)^2.$$

$$\sum_{k=1}^{n+1} (2k-1) = \sum_{k=1}^n (2k-1) + (2(n+1)-1) = n^2 + (2n+1) = (n+1)^2.$$

Poiché $P(1)$ è vera e $P(n) \Rightarrow P(n+1)$, la $P(n)$ è vera $\forall n \in N, n \geq 1$.



ATTIVITA' N. 5:

Dimostrare per induzione che $n^3 - n$ è divisibile per 3 per ogni n di N .

Indichiamo con $P(n)$ la proprietà da dimostrare.

$P(0)$ è ovvia. Dimostriamo che $P(n) \Rightarrow P(n+1)$.

$$(n+1)^3 - (n+1) = n^3 + 3n^2 + 2n + n - n = (n^3 - n) + 3(n^2 + n).$$

Il primo addendo dell'ultima espressione scritta, a causa dell'ipotesi $P(n)$, è divisibile per 3; anche il secondo addendo è ovviamente divisibile per 3. Dunque l'intera espressione è divisibile per 3, la $P(n+1)$ è vera e la proprietà è dimostrata.

La proprietà è facilmente dimostrabile anche senza utilizzare il principio di induzione.

La proprietà è ovviamente vera per $n=0$ ed $n=1$; supponiamo dunque che sia $n > 1$.

Dato che è $n^3 - n = (n-1)n(n+1)$, si tratta del prodotto di tre numeri naturali consecutivi, uno (ed uno solo) dei quali è certo divisibile per 3.³



ATTIVITA' N. 6:

Pierre Fermat, matematico francese del diciassettesimo secolo, formulò la congettura che tutti i numeri della forma $2^{2^n} + 1$ sono numeri primi.

³ Basta fare la "numerazione per tre",: 1,2,3,4,5,6,7,8,9,10,11,12

Utilizzare *DERIVE* per verificare questa congettura per valori di n da 1 a 6.

Ci si comporti in modo simile a quanto indicato nella Attività n. 1.

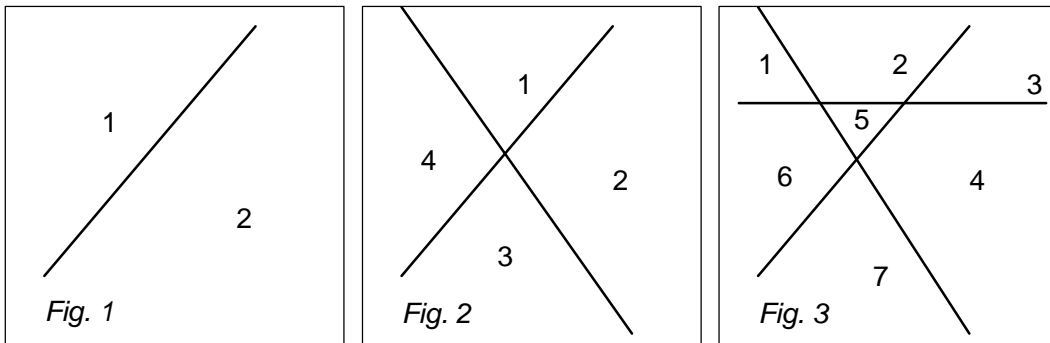
Può sorprendere il fatto che in pochi istanti si ottiene lo stesso risultato che il matematico svizzero Leonhard Euler (Eulero) ottenne solo un secolo dopo la congettura di Fermat, cioè che la congettura è falsa già per $n = 5$.



ATTIVITA' N. 7:

In quante regioni il piano risulta suddiviso da n rette a due a due intersecantesi?

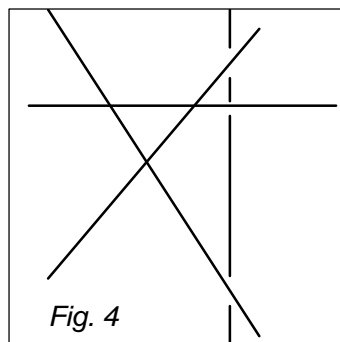
Iniziamo con l'osservare che per ipotesi le rette sono tutte distinte, non vi sono coppie di rette parallele e tre o più rette non possono passare per il medesimo punto.



Se indichiamo con $r(n)$ il numero delle regioni che cerchiamo, basta un'occhiata alle precedenti figure per poter affermare che $r(1) = 2$, $r(2) = 4$, $r(3) = 7$.

Ovviamente $r(0) = 1$: se non ho alcuna retta il piano viene "suddiviso" in un'unica regione: il piano stesso.

Un ragionamento di tipo induttivo ci permetterà di ottenere facilmente una relazione tra $r(n+1)$ ed $r(n)$: se inseriamo, ad esempio nella Fig. 3, una nuova retta che rispetta le ipotesi dette, questa intersecherà ciascuna delle tre rette preesistenti in un punto restando così suddivisa in 4 parti (due segmenti e due semirette, che in Fig. 4 sono state disegnate "distaccate" per meglio metterle in evidenza); ciascuna di questi segmenti o semirette divide a sua volta in due parti ciascuna delle 4 regioni che attraversa. Otterremo così in tutto $7 + 4 = 11$ regioni.



Più in generale una nuova retta intersecherà le n rette preesistenti in altrettanti punti restando così suddivisa in $n+1$ parti ($n-1$ segmenti e due semirette), ciascuna delle quali divide in due ogni regione che attraversa.

Avremo così:

$$r(n+1) = r(n) + n + 1 \quad (1).$$

Se applico la formula (1) al caso in cui inseriamo una n -esima retta dopo $n-1$ rette preesistenti avremo:

$$r(n) = r(n-1) + n \quad (2).$$

La formula (2) risponde alla domanda che ci eravamo posti. Tale formula, essendo ricorsiva, cioè esprimendo $r(n)$ in funzione di $r(n-1)$, nel caso volessimo calcolare ad esempio $r(10)$, ci costringe a calcolare anche $r(9)$, $r(8)$ e così via fino ad $r(0)$, che come è noto, vale 1.

Con un facile ragionamento possiamo però trasformare la formula (2) in una equivalente "forma chiusa" cioè in una formula che ci permette di esprimere direttamente $r(n)$ in funzione di n .

Applichiamo ripetutamente la formula (2):

$$\begin{aligned} r(0) &= 1, \\ r(1) &= r(0) + 1, \\ r(2) &= r(1) + 2, \\ &\dots\dots\dots \\ r(n-1) &= r(n-2) + n-1, \\ r(n) &= r(n-1) + n. \end{aligned}$$

Sommando membro a membro le precedenti uguaglianze avremo:

$$\begin{aligned} r(0) + r(1) + r(2) + \dots + r(n-1) + r(n) &= \\ = 1 + r(0) + 1 + r(1) + 2 + \dots + r(n-2) + n - 1 + r(n-1) + n. \end{aligned}$$

Semplificando i termini uguali a primo e a secondo membro avremo:

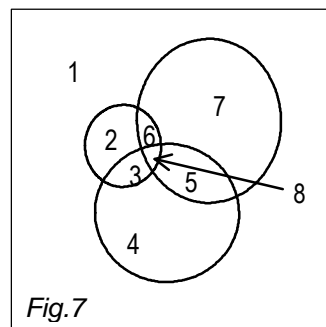
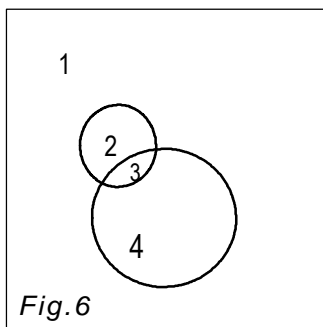
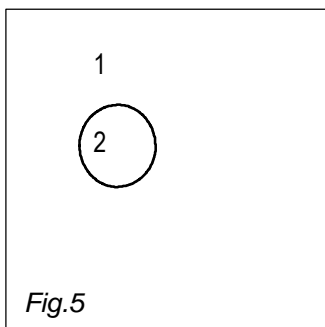
$$r(n) = 1 + [1 + 2 + \dots + (n-1) + n] = 1 + \frac{n(n+1)}{2}.$$

Abbiamo utilizzato la nota formula che esprime la somma dei primi n numeri naturali positivi (V. anche Attività N. 2).



ATTIVITA' N. 8:

In quante regioni n circonferenze distinte (che a due a due si intersecano) dividono il piano?



Il problema non è molto diverso dal precedente.

Indichiamo con $c(n)$ il numero delle regioni in cui viene diviso il piano da n circonferenze.

Ovviamente il piano viene diviso da una circonferenza in due regioni, una interna alla circonferenza e l'altra esterna ad essa (v. Fig. 5):

$$c(1) = 2.$$

Così il piano viene diviso in quattro regioni da due circonferenze che si intersecano a due a due (v. Fig. 6):

$$c(2) = 4.$$

Tre circonferenze a due a due secanti individuano otto regioni sul piano (v. Fig. 7):

$$c(3) = 8.$$

Si può così essere portati a pensare che la formula generale sia:

$$c(n) = 2^n.$$

Questa congettura però viene smentita dalla seguente osservazione: per ipotesi ogni nuova circonferenza che viene tracciata deve intersecare tutte le preesistenti circonferenze suddividendo ciascuna di esse in due regioni: quella interna e quella esterna alla nuova circonferenza. Così:

$$c(n + 1) = c(n) + 2n.$$

Avremo così:

$$\begin{aligned} c(1) &= 2 \\ c(2) &= c(1) + 2 \\ c(3) &= c(2) + 4 \\ c(4) &= c(3) + 6 \\ &\dots\dots\dots \\ c(n) &= c(n - 1) + 2(n - 1) \\ c(n + 1) &= c(n) + 2n. \end{aligned}$$

Sommiamo membro a membro:

$$\begin{aligned} c(1) + c(2) + c(3) + \dots + c(n) + c(n+1) &= \\ = 2 + c(1) + 2 + c(2) + 4 + \dots + c(n-1) + 2(n-1) + c(n) + 2n. \end{aligned}$$

da cui, semplificando tra i due membri:

$$\begin{aligned} c(n + 1) &= 2 + 2 + 4 + \dots + 2(n - 1) + 2n = \\ &= 2 + 2(1 + 2 + \dots + (n - 1) + n). \end{aligned}$$

Ma l'espressione tra parentesi tonda rappresenta la somma dei numeri naturali non maggiori di n ; come abbiamo già visto più volte, il suo valore è dato dalla nota formula:

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

Così:

$$c(n + 1) = 2 + n(n + 1).$$

Per ottenere una formula relativa al caso di n circonferenze anziché $n+1$, basterà sostituire $n - 1$ ad n nella relazione ora trovata:

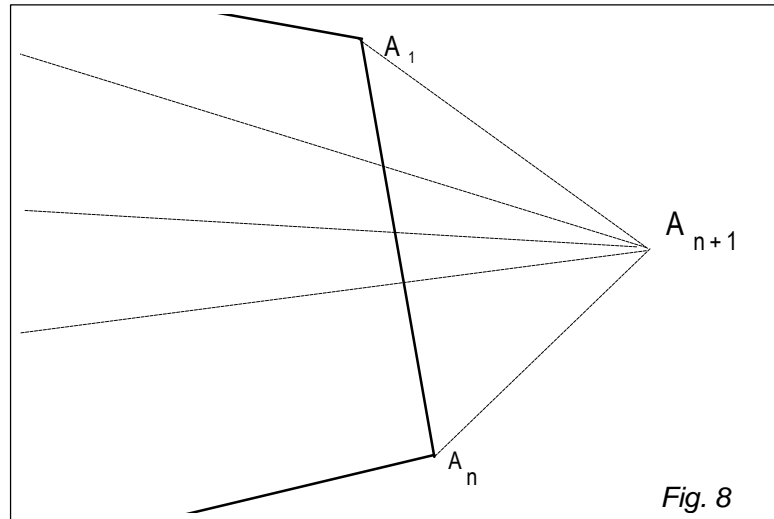
$$c(n) = 2 + (n - 1)n = n^2 - n + 2.$$

ATTIVITA' N. 9:



Determinare il numero $d(n)$ delle diagonali di un poligono (convesso) di n lati.

Si supponga di voler inserire un nuovo vertice "compreso" tra l'ultimo vertice A_n ed il primo A_1 di un poligono di n lati che ha i vertici A_1, A_2, \dots, A_n disposti, ad esempio, in senso antiorario.



Si osservi (v. Fig. n. 8) che, per ogni nuovo vertice che viene aggiunto al poligono, il numero delle diagonali aumenta di $n - 2$ (tutti i segmenti che congiungono il nuovo vertice con tutti i vertici preesistenti, tranne i due più vicini, A_1A_{n+1} ed A_nA_{n+1} , che sono lati del nuovo poligono); a questi va però aggiunto 1: infatti il segmento A_1A_n , che nel precedente poligono era lato, ora diviene diagonale.

Avremo:

$$\begin{aligned}
 d(3) &= 0 \\
 d(4) &= d(3) + 2 = 2 \\
 d(5) &= d(4) + 3 = 5 \\
 &\dots\dots\dots \\
 d(n+1) &= d(n) + n - 1
 \end{aligned}$$

Si deduca da questa, con il procedimento visto nelle precedenti Attività, la nota formula:

$$d(n) = \frac{n(n-3)}{2}.$$



ATTIVITA' N. 10:

Dati $n, m \in \mathbb{N}, m \neq 0$, dimostrare l'esistenza degli interi q ed r tali che

$$n = qm + r, \text{ con } 0 \leq r < m$$

procedendo per induzione rispetto ad n .⁴

In altre parole, qualunque siano i numeri naturali n ed m (il secondo diverso da zero), esistono sempre un quoziente ed un resto della divisione intera tra n ed m .

Consideriamo m fissato; dimostriamo $P(0)$, cioè vediamo che la proprietà è vera nel caso $n = 0$:

In tal caso q ed r esistono e sono entrambi nulli: infatti $0 = 0m + 0$. Si noti che $0 \leq r = 0 < m$, in virtù dell'ipotesi $m \in \mathbb{N}, m \neq 0$.

Supponiamo ora vera la $P(n)$ e deduciamo da essa la $P(n+1)$:

Poiché $P(n)$ è per ipotesi vera, esistono q ed r tali che

$$n = qm + r, \text{ con } 0 \leq r < m.$$

Aggiungo 1 membro a membro:

$$n + 1 = qm + (r + 1).$$

Quindi la proprietà $P(n+1)$ è vera, se si suppone $r + 1 < m$.

⁴ Vedi es. 1.6-1 in: G. C. Barozzi, "Primo Corso di Analisi Matematica" - Ed. Zanichelli.

Se invece $r + 1 = m$, allora:

$$n + 1 = qm + (r + 1) = qm + m = (q + 1)m + 0$$

e anche in questo caso è assicurata l'esistenza del quoziente $(q+1)$ e del resto (che è $< m$, essendo 0).

Si noti che non può essere $r + 1 > m$: se così fosse, infatti, sarebbe $r > m - 1$, ossia $r \geq m$ contrariamente a quanto affermato dall'ipotesi $P(n)$.



ATTIVITA' N. 11:

Definire ricorsivamente con *DERIVE* la funzione potenza a^n , con $a, n \in N, a > 0$.

In effetti non sarebbe necessario: *DERIVE* è già fornito dell'operazione di elevamento a potenza, rappresentato dal simbolo di accento circonflesso $^$. Vogliamo ugualmente creare questa funzione potenza come esercizio di funzione ricorsiva.

Selezionare **Author**. Digitare la seguente funzione: **pot(a,n):=if(n=0,1,pot(a,n-1)a)**.

La funzione potenza viene così definita come = 1, nel caso che l'esponente sia 0, e come $a^{n-1}a$, in tutti gli altri casi.

Più in generale la funzione **if** supporta tre argomenti: il primo riporta una *condizione*, cioè una proposizione ⁵, il secondo rappresenta il valore da assegnare alla funzione nel caso in cui la condizione sia vera, il terzo il valore nel caso che la condizione sia falsa.

Digitare **pot(3,2) <↵>**, poi selezionare **Simplify**.

Viene così calcolata la potenza 3^2 .



ATTIVITA' N. 12:

Definire ricorsivamente con *DERIVE* una funzione, che potremmo chiamare FATT(n), che calcoli il fattoriale del suo argomento. ⁶

Si ricordi che :

$$n! := \begin{cases} 1, & \text{se } n = 0 \\ (n - 1)!n, & \text{altrimenti.} \end{cases}$$



ATTIVITA' N. 13:

La *progressione aritmetica* di primo elemento a e ragione d può essere così definita ricorsivamente:⁷

$$a_n := \begin{cases} a, & \text{se } n = 0 \\ a_{n-1} + d, & \text{altrimenti.} \end{cases}$$

Dimostrare che $a_n = a + nd$, per ogni naturale n .

$$\begin{aligned} a_0 &= a, \\ a_1 &= a_0 + d, \\ a_2 &= a_1 + d, \\ &\dots\dots\dots \\ a_n &= a_{n-1} + d. \end{aligned}$$

Sommando membro a membro le precedenti uguaglianze e semplificando gli addendi uguali dei due membri si ottiene:

$$a_n = a + d + d + \dots + d = a + nd.$$

Dimostrare che la somma dei termini di indici non superiori ad n è data dalla formula:

⁵ o, meglio, una *forma proposizionale*.

⁶ In effetti anche questa funzione non sarebbe necessaria: *DERIVE* è già fornito dell'operatore di fattoriale, rappresentato dal consueto simbolo **!**. Si noti che anche usando la funzione predefinita, il risultato di $0!$ è 1.

⁷ V. G.C. Barozzi, *op. cit.*, es. 1.6-6.

$$(n+1)\left(a + \frac{nd}{2}\right).$$

Sommando membro a membro le uguaglianze della precedente dimostrazione e utilizzando la formula prima dimostrata, si ottiene:

$$a_0 + a_1 + a_2 + \dots + a_n = a + (a+d) + (a+2d) + \dots + (a+nd)$$

ovvero, in simboli:

$$\sum_{k=0}^n a_k = \sum_{k=0}^n (a + kd).$$

Applicando le proprietà commutativa ed associativa della somma, si ottiene facilmente:

$$(n+1)a + d + 2d + \dots + nd = (n+1)a + d(1 + 2 + \dots + n) =$$

per la solita formula:

$$= (n+1)a + d \frac{n(n+1)}{2} = r \text{ accogliendo } (n+1) = (n+1)\left(a + \frac{nd}{2}\right).$$



ATTIVITA' N. 14:

I cosiddetti "numeri di Fibonacci" sono definiti ricorsivamente come segue: ⁸

$$F_0 := 0, F_1 := 1, F_n := F_{n-1} + F_{n-2}, \text{ con } n \geq 2.$$

Calcolare con *DERIVE* i primi otto numeri di Fibonacci.

In questo caso useremo due istruzioni **if** "innestate" l'una dentro l'altra per prendere in considerazione i tre casi possibili: $n = 0, n = 1, n \geq 2$.

Selezionare **Author**.

Digitare **fib(n):=if(n=0,0,if(n=1,1,fib(n-1)+fib(n-2))) <↵>**.

Questo equivale al seguente algoritmo:

```

se n=0
  allora fib(n)=0
altrimenti se n=1
  allora fib(n)=1
altrimenti fib(n)=fib(n-1)+fib(n-2).

```

Per calcolare i primi otto numeri di Fibonacci si consiglia di utilizzare la ben nota funzione **vector**.

Un altro modo, più astuto, di definire la funzione **fib(n)** con l'uso di una sola istruzione **if** è il seguente:

fib(n):=if(n<=1,n,fib(n-1)+fib(n-2)) <↵>.

Si noti che l'operatore relazionale \leq viene digitato con \leq ma viene poi indicato con il simbolo usuale nella finestra di Algebra.



fibon

SINTESI

MENU

Selezionando **Calculus Sum** si ottiene il calcolo della somma di una serie o di una sommatoria.

EDITING

Per assegnare un valore ad una variabile, selezionare **Author**, poi digitare il nome della variabile seguito dai caratteri **:=** ed infine il valore desiderato (es.: **x:=4**).

Per definire una funzione senza passare attraverso il menu **Declare Function**, selezionare **Author** e digitare il nome della funzione, ad esempio **F(x)**, seguito da **:=** ed infine l'espressione desiderata.

L'operatore relazionale \leq si ottiene digitando **<=**.

FUNZIONI

La funzione **vector(espressione,variabile,vi,vf,passo)** costruisce un vettore formato dai valori di **espressione** al variare di **variabile** dal valore iniziale **vi** al valore finale **vf**, con incremento = **passo**. Se **passo** è omissso, viene assunto **passo = 1**.

Il comando **vector** realizza in pratica una iterazione enumerativa simile al ciclo *for*.... esistente nei linguaggi di programmazione.

La funzione **if(condizione,espressione1, espressione2)** fornisce l'**espressione1** se **condizione** è vera, l'**espressione2** se **condizione** è falsa.