

# PROGETTO 1

Scheda progetto	
<b>Nome progetto</b>	Somma - prodotto
<b>Problema</b>	Calcolo della somma e del prodotto di due numeri
<b>Algoritmo creato</b>	Somma e prodotto tra due numeri
<b>Teoria di Pascal</b>	Struttura dei programmi in Pascal

Partiamo con calma, siccome questo è il primo progetto in Pascal...

Questo programma, così come tutti gli altri, è strutturato in modo da far ragionare chi lo legge, in modo che il programma in Pascal sia solo il risultato di un lavoro algoritmico su carta fatto in precedenza.

Dopo l'algoritmo troverete il codice del programma in Pascal e, di seguito, tutta la descrizione e le eventuali parti teoriche.

## **Analisi del problema**

In questo caso non si deve fare una vera analisi matematica.

Si deve in generale calcolare il prodotto e la somma di due numeri qualsiasi.

Costruiamo una tabella di valori per verificare il nostro metodo.

<b>A</b>	<b>B</b>	<b>A+B</b>	<b>A • B</b>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1
2	3	5	6
3	4	7	12
4	5	9	20
10	5	15	50

In ogni tabella di calcolo come questa, troverete le combinazioni con 0 e 1, in quanto, a livello matematico, questi due numeri possono influire in modo negativo su tutto il programma. Basti pensare a una divisione o a una potenza...

La tabella, in fondo, ci serve appunto a verificare i casi in cui il nostro ragionamento non porterebbe a un risultato in campo reale.

In questa tabella non esistono soluzioni impossibili, che potrebbero causare un mal funzionamento del programma.

## **Creazione dell'algoritmo**

Proviamo a pensare a quelle che sarà necessario calcolare, leggere da tastiera e scrivere.

Dovremo innanzitutto leggere da tastiera due variabili, quindi elaborare la loro somma e il loro prodotto e stamparli su schermo.

1. Leggi A
2. Leggi B
3. Scrivi A+B
4. Scrivi A•B

## Creazione del programma Pascal

Ecco di seguito il codice per creare il programma in Pascal. Visto che è la prima volta che vedete un programma, non spaventatevi: cercate di comprendere ciò che vedete scritto in modo globale, poi passate all'analisi seguente.

```
program som_pro;
var a,b: integer;
begin
readln (a);
readln (b);
writeln ('La soma è' , a+b);
writeln ('Il prodotto è' , a*b);
end.
```

## Analisi del programma

Essendo questo il primo programma in Pascal creato, andremo ad analizzare ogni riga singolarmente, e studieremo ogni singolo comando. In seguito, il tutto avverrà in modo più generico.

```
program som_pro;
```

Questa riga è in genere la prima a essere posta all'inizio di ogni programma in Pascal. Si tratta di un modo come un altro per dare un nome identificativo al programma che si sta creando. Il nome del programma deve iniziare con una lettera e può contenere, in seguito:

- Lettere
- Cifre
- Segni di sottolineatura (\_)

Non sono ammessi spazi.

Da notare la struttura

```
program nomeprogramma ;
```

In pratica:

1. Inserire il termine privato "program"
2. Inserire il nome del programma
3. Inserire un punto e virgola (;)

Questa fase in cui si indica il nome del programma è detta **intestazione del programma**.

```
var a,b: integer;
```

Questa riga è detta riga di **dichiarazione**.

Qui si specificano tutte le variabili e tutte le costanti che saranno utilizzate durante il programma: da notare che se non sono specificate, le variabili non possono essere usate.

Per dichiarare delle variabili si usa lo schema

```
var var1, var2, var3, varx : tipovariabili;
    vara, varb, varc, varz : altrotipovariabili;
```

Quindi lo schema prevede:

1. Inserire il termine riservate "var"
2. Inserire i nomi di tutte le variabili separate da virgole
3. Inserire due punti (:)
4. Specificare il tipo di quelle variabili
5. Inserire un punto e virgola

Da notare che esistono molti tipi di variabili, numerici, alfabetici, che saranno studiati in dettaglio in progetti seguenti.

Nel nostro caso è stato usato il tipo **integer**, ossia variabili a cui è possibile assegnare valori numerici interi positivi o negativi compresi entro certi limiti specificati.

E' possibile anche specificate delle costanti, utilizzando uno schema molto semplice

```
const a = 0,11;  
const b = 'Maria';
```

Le costanti hanno uno schema diverso:

1. Inserire il termine riservato "const"
2. Inserire il nome della costante
3. Inserire un uguale (=)
4. Inserire il valore della costante: se si tratta di un valore testuale (**stringa**), bisogna inserirlo tra apici singoli

**begin**

Questa è una delle parole riservate del pascal più importanti di tutto il linguaggio: indica l'inizio del programma, ed è posto subito dopo le dichiarazioni. Il termine **begin** sarà usato anche in altri casi, come vedremo in futuro.

```
readln (x);  
writeln (x);
```

Dopo il termine **begin** inizia tutta la lunga serie di **istruzioni**, in cui si lavora con le variabili e le costanti dichiarate in precedenza.

I termini **readln** e **writeln** sono termini riservati con due compiti ben precisi:

- **readln**: legge dei valori inseriti da tastiera. In questo comando, dopo la lettura, il puntatore viene spostato su una nuova riga. Se si vuole continuare sulla stessa bisogna utilizzare il comando più semplice **read**.
- **writeln**: questo comando stampa su schermo i valori di una o più variabili. E' possibile aggiungere del testo, ma deve essere compreso tra apici singoli. In questo caso, dopo aver scritto, il puntatore viene portato a capo. Per rimanere sempre nella stessa riga si utilizza il comando più semplice **write**.

La struttura di un'istruzione è, in genere, questa:

```
nomeistruzione (vara, varb, varc);
```

Ossia:

1. Inserire il nome dell'istruzione (es. read, write)
2. Aprire una parentesi tonda
3. Inserire le variabili, le costanti o le stringhe su cui eseguire l'istruzione, separate da una virgola
4. Chiudere la parentesi tonda
5. Mettere un punto e virgola

Un esempio di istruzione potrebbe essere

```
readln (a,b);  
writeln ('Hai inserito i numeri' , a , 'e' , b);
```

Quando si lavora con i dati, è possibile elaborarli, eseguendo dei calcoli matematici. In questo caso si utilizzano dei simboli convenzionali del Pascal. Abbiamo visto "+" e "\*", usati per addizione e moltiplicazione. Possiamo già introdurre il "-", usato per le sottrazioni.

Al termine delle istruzioni e quindi del programma intero, si pone la scritta **end** seguita da un punto.

**end.**

La struttura per chiudere un programma è:

1. Inserire la parola riservata "end"
2. Inserire un punto

Abbiamo a questo punto finito di commentare questo programma, che servirà da base per ogni programma: sono state presentate le principali regole di sintassi, importantissime sempre (quali i punti e virgola alla fine di ogni frase, il punto alla fine del programma, la virgola negli elenchi...).

### Approfondimento 1: Abbellire il programma

Abbiamo a questo punto creato il nostro primo programma in Pascal e siamo stati in grado di capire come funziona in generale il meccanismo di questo linguaggio di programmazione. Il nostro programma, tuttavia, oltre ad essere veramente ridicolo e banale, è del tutto privo di grafica. Quando un soggetto andrà ad aprire il nostro file, si troverà un puntatore lampeggiare, e sarà lui **da solo** a dover capire che è necessario inserire dei dati. Questo problema può essere tuttavia facilmente risolto utilizzando la parola riservata "writeln". Ecco come possiamo migliorare di molto il nostro programma...

```
program som_pro;
var a,b: integer;
begin
writeln ('Questo programma calcola somma e prodotto di due
  numeri interi');
writeln ('Inserisci due numeri interi');
write ('a=      ');
readln (a);
write ('b=      ');
readln (b);
writeln ('La somma è      ', a+b);
writeln ('Il prodotto è    ', a*b);
readln;
end.
```

Decisamente più bello e comprensibile, vero? Sicuramente vi sarete chiesti a cosa serve quel "readln;" alla fine del programma. Bene, ne parleremo diffusamente nell'approfondimento 3.

### Approfondimento 2: Matematica e simboli

Abbiamo imparato a lavorare con i numeri, in particolare con addizione e moltiplicazione. Abbiamo accennato inoltre la sottrazione, che si comporta in modo del tutto analogo agli altri operatori, solo che si indica con il simbolo "-". Per il momento non parliamo nemmeno di divisione perché in Pascal il discorso è leggermente più complesso di quello che può sembrare. Comunque, avendo già introdotto tre simboli matematici, è necessario parlare anche di precedenze: come tutti sappiamo, si eseguono prima le moltiplicazioni (e le divisioni) e poi le addizioni e le sottrazioni.

$$2 + 3 \cdot 2 = 2 + 6 = 8$$

Ma, se vi sono delle parentesi, queste prevalgono sui simboli in generale:

$$\begin{aligned}(2+3) \cdot 2 &= 6 \cdot 2 = 12 \\ [2 + (3 \cdot 4) \cdot 3] &= [2 + 12 \cdot 3] = [2 + 36] = 38 \\ \{2 + [(5 - 3) \cdot 2]\} &= \{2 + [2 \cdot 2]\} = \{2 + 4\} = 6\end{aligned}$$

In Pascal, come del resto in tutti i linguaggi ad alto livello, si possono utilizzare le parentesi, **ma nelle precedenze matematiche si utilizzano solo quelle tonde ( )**. Le quadre e le graffe hanno infatti altre funzioni, di cui si parlerà molto più avanti. Riprendendo l'esempio di prima, l'operazione in Pascal diverrebbe...

$$(2 + ((5-3) \cdot 2)) = (2 + (2 \cdot 2)) = (2 + 4) = 6$$

Quando si lavora con espressioni molto lunghe, è importante prestare attenzione, in quanto ogni parentesi aperta deve necessariamente essere chiusa.

### Approfondimento 3: Attesa

Se avete provato il primo programma di questo progetto, dovrete aver notato che i risultati volavano via velocissimi, era quasi impossibile leggerli.

Ebbene, per ovviare a questo inconveniente, è possibile procedere in due modi.

Il primo consiste nell'inserire, prima della fine del programma, l'istruzione

**readln;**

In questo modo il computer aspetta di leggere qualcosa da tastiera, prima di eseguire il comando successivo (ossia la fine del programma).

Il secondo metodo, leggermente più lungo, è composto da due righe di codice:

1. Inserire, dopo l'intestazione del programma, la scritta  
**uses crt;**
2. Inserire, prima della fine del programma, la scritta  
**repeat until** keypressed;  
ossia ripeti finché non viene premuto un tasto.

Il secondo metodo, anche se più lungo, è decisamente più consigliato.

### Proposte

In questa parte vengono proposte delle tracce per creare dei programmi in Pascal. Tutti sono basati sulla costruzione di algoritmi (per ora molto semplici) e sulla scrittura di un programma in Pascal. Non si tratta di veri e propri esercizi, ma di puri spunti per la creazione di programmi in linguaggio.

1. Dati due numeri, stampare su schermo il doppio della loro somma e il doppio del loro prodotto.
2. Dati due numeri, stampare su schermo la loro differenza, la loro somma, il loro prodotto e i loro quadrati.
3. Dati due numeri, stampare su schermo la somma di:
  - a. Il quadrato del primo
  - b. Il quadrato del secondo
  - c. Il doppio prodotto dei due numeriSi tratta in pratica di calcolare il quadrato del binomio.
4. Dati tre numeri, stampare su schermo il risultato del quadrato del trinomio, ricordando che  $(a + b + c)^2 = a^2 + b^2 + c^2 + 2ab + 2bc + 2ac$ . Per eseguire il programma bisogna quindi calcolare i valori dei sei monomi del prodotto notevole e sommarli insieme.
5. Dati due numeri, stampare su schermo il risultato del cubo del binomio, ricordando che  $(a + b)^3 = a^3 + b^3 + 3a^2b + 3ab^2$ .
6. Dati due numeri, stampare su schermo il prodotto della somma e della differenza, senza usare il prodotto notevole. Calcolare quindi  $(a+b)(a-b)$
7. Eseguire l'esercizio 6 calcolando però il prodotto notevole:  $(a+b)(a-b) = a^2 - b^2$

## Riepilogo

Simboli nuovi	
+	Addizione
-	Sottrazione
*	Moltiplicazione
(	Precedenze matematiche
)	Precedenze matematiche

Punteggiatura nuova	
:	Specificazione tipo variabili
,	Separazione negli elenchi
;	Termine di ogni istruzione
.	Termine del programma

Termini riservati nuovi	
<b>begin</b>	Inizio del programma
<b>const</b>	Dichiarazione delle costanti
<b>end</b>	Fine del programma
<b>integer</b>	Variabili di tipo intero
<b>read</b>	Leggi
<b>readln</b>	Leggi e vai a capo
<b>program</b>	Indica il nome del programma
<b>var</b>	Dichiarazione delle variabili
<b>write</b>	Scrivi
<b>writeln</b>	Scrivi e vai a capo

Progetto creato da Squall1988

<http://www.pslife.net>

© PiEsseLife - 2003