

### 3 - a) I PRINCIPALI TIPI DI VARIABILI NUMERICHE b) LE VARIABILI "STRINGA"

#### a) Alcuni fra i principali tipi di variabili numeriche

**FREE PASCAL mette a disposizione parecchi tipi di variabili *interi* e *non interi*.**

Ma lasciando i tanti possibili approfondimenti all'eventuale iniziativa del lettore, che potrà trovarli ad es. sul sito [www.freepascal.org](http://www.freepascal.org), diciamo che, avendo necessità di utilizzare una variabile di tipo intero, la si dichiarerà in genere come **VAR integer** oppure come **VAR longint**, tenendo presente che

- ♪ una variabile di tipo **integer** può assumere i suoi valori nell'intervallo da **-32768** a **+32767**
- ♪ e una variabile di tipo **longint** da **-2147483648** a **+2147483647**.

♥ **E' di estrema importanza tener conto dell'intervallo di variabilità (range).**

**Se ad esempio una variabile n in un dato programma PASCAL è stata dichiarata di tipo integer, e in quel programma compare l'istruzione `n:=40000`,**

**oppure la coppia di istruzioni successive `n:=32767; n:=n+1`, allora sarà un bel guaio!!!**

**In fase di compilazione o di esecuzione verrà segnalato un errore di "traboccamento" (*overflow*).**

Una variabile numerica a valori **non necessariamente interi** in generale è dichiarata come **VAR real**.

Le variabili di tipo *real* hanno un *range* che copre, perlomeno, l'intervallo da  $1.5 \cdot 10^{-45}$  a  $3.4 \cdot 10^{38}$  (nel caso del sottotipo *single*); gli altri sottotipi *double* e *extended* hanno un *range* molto più ampio.

Il numero di byte occupati in memoria può essere di 4, di 8, o di 10. Il programmatore potrà specificare il sottotipo con una dichiarazione come *VAR x: double*, oppure non specificarlo scrivendo *VAR x: real*; in tal caso, la variabile in gioco è destinata a diventare una *single* o una *double* a seconda del processore. Consulta [www.freepascal.org](http://www.freepascal.org) per informazioni più dettagliate; vedi anche il paragrafo 7b a pag. 11.

♥ **OCCHIO!** - In PASCAL il **separatore** della parte intera dalla decimale è il **PUNTO** e non la virgola.

- Si ha la possibilità, quando una *write* o una *writeln* è riferita ad un numero di tipo intero, di **far sì che il numero occupi, sul monitor, tanti spazi quanti esattamente vogliamo noi**.
  - Ad esempio, se *prodotto* è una variabile *integer* o *longint*, l'istruzione **`write (prodotto:8)`** fa sì che sul monitor compaia il valore che in quel momento la variabile *prodotto* possiede, scritto in modo da occupare **esattamente 8 posizioni di carattere, e allineato a destra** in tale spazio. In questo contesto, il numero 8 viene detto "**ampiezza**".
- **Se noi mandiamo in output, tramite una *write* o una *writeln*, un numero di tipo *real*, esso ci apparirà sul monitor in "notazione esponenziale"**. Vale a dire, vedremo il numero scritto come **prodotto di un fattore compreso fra 1 (incluso) e 10 (escluso), per una opportuna potenza di 10**.
  - Esempio. Supponiamo che in un dato istante la variabile *x*, di tipo *real*, abbia il valore **123.45**. Allora l'istruzione **`write (x)`** farebbe comparire sul monitor **1.2345000000000000E+002** (che significa 1.2345 moltiplicato per  $10^2$ ; "**E**" sta per "**esponente di 10**").

**Se vogliamo invece che l'output appaia scritto in notazione *non* esponenziale**, dovremo integrare l'istruzione *write* come nell'esempio che segue: **`write (x:20:12)`**. Tale istruzione avrebbe l'effetto di far scrivere sul monitor il valore della variabile *real* *x*, scritto in notazione **NON** esponenziale, in modo da occupare un campo di esattamente **20 caratteri, di cui 12 riservati alle cifre dopo il punto decimale**.

➤ Ad esempio, nel caso **`x=123.45`**, si avrebbe

**123.45000000000000** (4 spazi vuoti all'inizio: numero *allineato a destra* in un campo di 20 caratteri)

NOTA 1 - Lo ribadiamo: i numeri non interi hanno sempre, in PASCAL, **il PUNTO e non la virgola come separatore per le cifre decimali**. Anche quando un non-intero viene inserito in *input*, in fase di esecuzione di un programma scritto in PASCAL, occorre regolarsi così.

NOTA 2 - L'aggettivo *reale* è un po' "sprecato" in questo contesto. Sì, è vero, si tratta di numeri reali, però, per il fatto di non poter avere infinite cifre decimali, sono senz'altro addirittura *razionali*.

#### b) Cenni alle variabili "stringa"

##### **STRINGA = SEQUENZA DI CARATTERI**

```
program esempiosullevariabilistringa; uses crt;
var nome: string [20];
begin
  clrscr;
  writeln ('Come ti chiami?');
  readln (nome);
  if nome='Mario' then writeln ('Ti chiami come me!')
    else writeln ('Buona giornata ', nome);
  readln;
end.
```

##### **var nome: string [20]**

"nome" è una **variabile stringa**.

La "scatoletta" chiamata "nome" non è destinata a contenere un numero, bensì una "**stringa**", ossia una *sequenza di caratteri*.

Il numero [20] *entro parentesi quadre* indica che la stringa potrà avere una *lunghezza massima* di 20 caratteri.

Il caso particolare *string* [1] può essere rimpiazzato dal tipo "**carattere**" (*char*).

Esempio: **var consonante: char**