I SOTTOPROGRAMMI RICORSIVI

Se si stabilisce che un sottoprogramma può chiamare qualsiasi altro sottoprogramma (ad esso accessibile) dobbiamo esaminare la possibilità che richiami *se stesso*.

Quando ciò avviene si parla di *programmazione ricorsiva* e l'azione o la denotazione del richiamo viene detta **ricorsione.**

L'algoritmo di Euclide per trovare il MCD fra *m* e *n* con (*m* > *n)* potrebbe essere espresso come segue a parole come segue:

{per trovare MCD(m, n) si effettui la divisione fra m ed n. Sia r il resto della divisione. Se r = 0 allora n è il MCD, altrimenti il MCD(m, n) è uguale al MCD(n, r)} cioè:

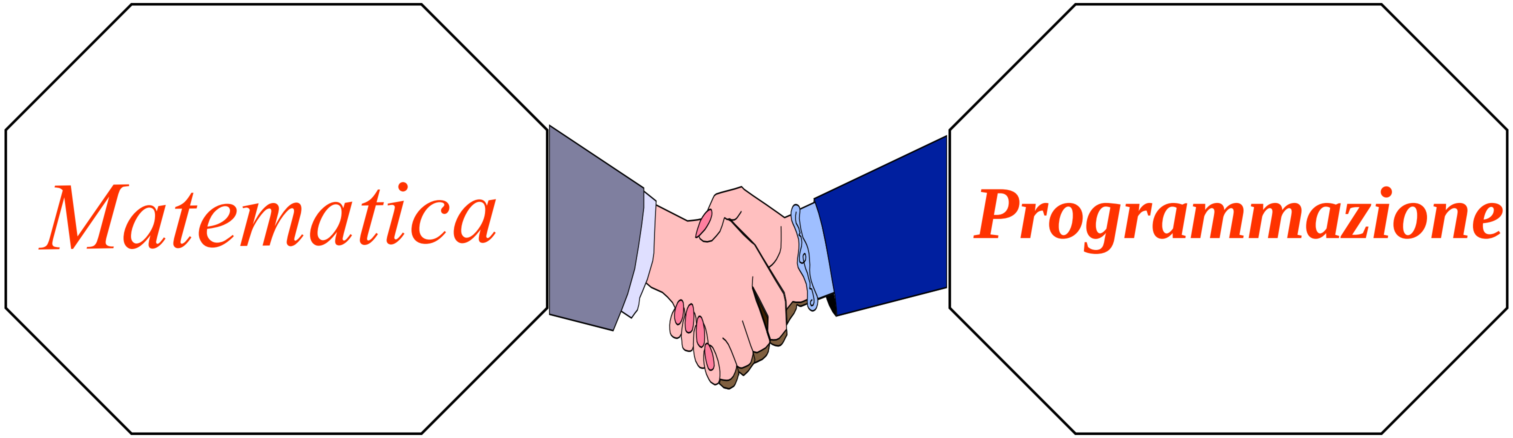
MCD(15, 9) = MCD(9, 6) = MCD(6, 3) = 3

certamente si ricorre ma su argomenti sempre più piccoli fino a trovare un caso in cui non si ricorre più.

Definizione ricorsiva della funzione per il calcolo del MCD

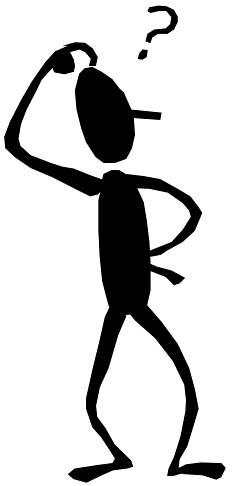
```
function MCD (M,N){
   var  R;
   R  = M%N;
   if (R==0) return N;
   else  return MCD(N, R);
}
```

chiamata ricorsiva



Costruire un sottoprogramma ricorsivo per un dato problema corrisponde a dare una definizione per *induzione* della sua soluzione.

Affinché un sottoprogramma ricorsivo sia correttamente concepito (in pratica perché non vada in loop infinito):



a) deve esserci un modo per uscire dal sottoprogramma evitando ogni chiamata ricorsiva (*corrisponde alla base di induzione*).

b) deve esserci almeno una chiamata ricorsiva con argomenti orientati alla soluzione di un sottoproblema più piccolo dell'originale (*corrisponde all'ipotesi d'induzione*).

Tutti i linguaggi di script consentono la ricorsione, perché tutti consentono una *gestione dinamica* degli ambienti locali.

Una chiamata ricorsiva comporta (come per una chiamata non ricorsiva) il salvataggio delle informazioni necessarie a riprendere successivamente il calcolo, e cioè:

- *il punto di ritorno*

- *i parametri attuali*

- *lo stato dell'ambiente locale.*

Funzione ricorsiva che calcola il fattoriale di un numero

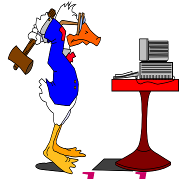
```
function fatt(N){
If (N==0)
   fatt=1;
else
   return N*fatt(N-1);
}
```

Tracing per N=3

fatt(3)=3*fatt(2)
fatt(2)=2*fatt(1)
fatt(1)=1*fatt(0)
fatt(0)=1

Quindi ricostruire la pila.
L’output sarà 6

Esercizio n. 6



1) *scrivere una function ricorsiva per il calcolo del fattoriale di un numero N >= 0 (che succede se N < 0?);*

2) *scrivere una function ricorsiva per il calcolo del numero di combinazioni di n oggetti presi a K a K; cioè C n,k*

3) *scrivere una procedura ricorsiva per eseguire a ricerca binaria su un vettore A(1 to N). Restituire l'indice dell'elemento cercato (se esiste) e l'esito della ricerca;*

4) *cosa calcola la seguente funzione ricorsiva?*

function F (ByVal N, K)

if (N <=0) F = 0 else return N ^ K + F(N - 1, K)