

Ricorsione

Ricorsione

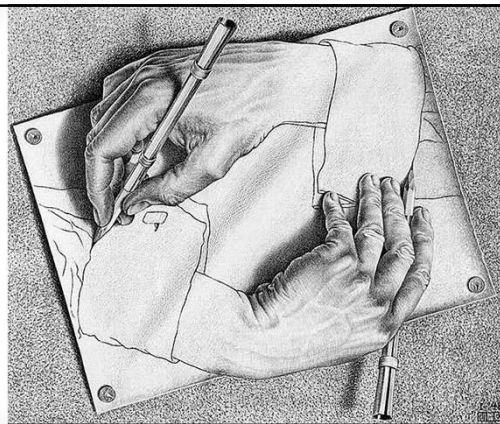
Procedure e funzioni ricorsive

Definizioni

- Un oggetto si dice **ricorsivo** se è definito totalmente o parzialmente in termini di sé stesso
- La ricorsione è un mezzo molto potente per le definizioni e le dimostrazioni matematiche (induzione)
- Si usano algoritmi ricorsivi quando il problema da risolvere presenta caratteristiche proprie di ricorsività (può essere risolto in termini di uno o più problemi analoghi ma di dimensioni inferiori)



- Sullo schermo del computer compare un computer sul cui schermo compare un computer sul cui schermo compare un computer e così via



Definizioni ricorsive

- Definizione dei numeri naturali:
 - 1) 1 è un numero naturale
 - 2) il successore di un numero naturale è un numero naturale
- Definizione di fattoriale:
 - 1) $0! = 1$
 - 2) se $N > 0$ allora $N! = N * (N-1)!$
- Calcolo del MCD tra due numeri A e B ($A > B$)
 - 1) dividere A per B
 - 2) se il resto R è zero allora $MCD(A,B)=B$ altrimenti $MCD(A,B)=MCD(B,R)$

Terminazione

- Il potere della ricorsività consiste nella possibilità di definire un insieme anche infinito di oggetti con un numero finito di comandi
- Il problema principale quando si usano algoritmi ricorsivi è quello di garantire una terminazione (caso terminale, condizione di fine, condizione iniziale)
- Non è sufficiente inserire una condizione di terminazione, ma è necessario che le chiamate ricorsive siano tali da determinare il verificarsi di tale condizione in un numero finito di passi

Ricorsione

Procedure e funzioni ricorsive

- Un sottoprogramma ricorsivo è una procedura (o funzione) all'interno della quale è presente una chiamata a se stessa o ad altro sottoprogramma che la richiama
- La ricorsione è **diretta** se la chiamata è interna al sottoprogramma altrimenti si dice **indiretta**.

Funzione ricorsiva per il MCD

```
int MCD(int A,B) {
    int R;
    //calcolo del resto della divisione tra A e B
    R= A % B;
    //calcolo di MCD
    If (R==0) Condizione di terminazione
        return B;
    else
        return(MCD(B,R));
}
```

Stampa invertita delle cifre di un numero

Procedura ricorsiva

```
void Inverti_Num(int N) {
    cout<< (N % 10);
    If ((N / 10) != 0)
        Inverti_Num(N / 10);
}

{M A I N}
...
Inverti_Num(425);
```

Procedura iterativa

```
void Inverti_Num(int N) {
    while (N / 10 != 0) do {
        cout<< (N % 10);
        N=N / 10;
    }
}

{M A I N}
...
Inverti_Num(425);
```

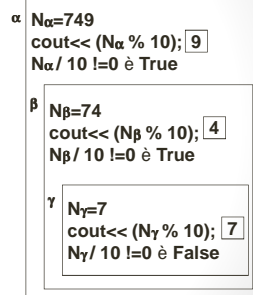
Istanziamento

- Ogni nuova chiamata di un sottoprogramma ricorsivo determina una nuova istanza dell'ambiente locale (distinto da quello precedente che comunque resta attivo).
- Ad ogni chiamata si alloca nuova memoria e questo può determinare problemi di spazio.
- I vari ambienti vengono salvati in una struttura di tipo LIFO (Stack o Pila) in modo che alla terminazione di una determinata istanza venga riattivata quella immediatamente precedente e così via

Diagramma di traccia Inverti

```
void Inverti_Num(int N) {
    cout<< (N % 10);
    If ((N / 10) != 0)
        Inverti(N / 10);
}

{M A I N}
...
Inverti_Num(749);
```



Fattoriale

```
int Fatt(int N {
    If N==0 return 1
    else return (N*Fatt(N-1));
};

{M A I N}
...
cout<<("Digita un valore : ");
cin>>N;
cout<<("Il fattoriale di „N,“ è „Fatt(N);
...

```

Ricorsione

Fattoriale
di 3

```
Nα=3
(Nα=0) → False
Fatt = Nα * Fatt(Nα-1) = 3 * Fatt(2)

  α
    Nβ=2
    (Nβ=0) → False
    Fatt = Nβ * Fatt(Nβ-1) = 2 * Fatt(1)

      β
        Nγ= 1
        (Nγ=0) → False
        Fatt = Nγ * Fatt(Nγ-1) = 1 * Fatt(0)

          γ
            Nδ= 0
            (Nδ=0) → True
            Fatt = 1

            Fatt = 1 * 1 = 1
          Fatt = 2 * 1 = 2
        Fatt = 3 * 2 = 6
```