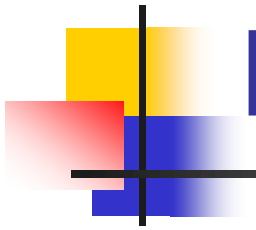


Ricorsione

Concetti chiave

- Ricorsione
- Realizzare funzioni ricorsive
- Approfondimenti sul passaggio dei parametri e valore restituito da una funzione
- Esercizio: Fattoriale
- Esercizio: Ricerca Binaria

Ricorsione

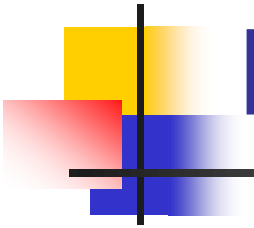
- Una funzione *ricorsiva* chiama se stessa:

```
void prova(void){
        …….
        prova();
        ……..
}
```

- E' una tecnica di programmazione che si applica a problemi che richiedono una soluzione ricorsiva, ossia espressa in termini di se stessa

  - Esempio: calcolo del fattoriale $n!$ Il fattoriale prevede una soluzione ricorsiva per definizione:

$$n! = n \cdot (n - 1)!$$



Ricorsione

- Come realizzare una funzione ricorsiva ?
- Inserire una o più istruzione if-else per prevedere i casi in cui la ricorsione termina
- Nei casi in cui la ricorsione deve continuare, si deve inserire la chiamata alla stessa funzione con gli opportuni parametri attuali
  - **Attenzione: se la funzione torna un valore la chiamata ricorsiva deve essere preceduta da return**

```
void prova(parametri formali){
      …….
      if (condizione) prova(parametri attuali);
}
```

```
int prova(parametri formali){
      …….
      if (condizione) return prova(par.attuali);
      else return valore;
}
```

Ricorsione: Fattoriale

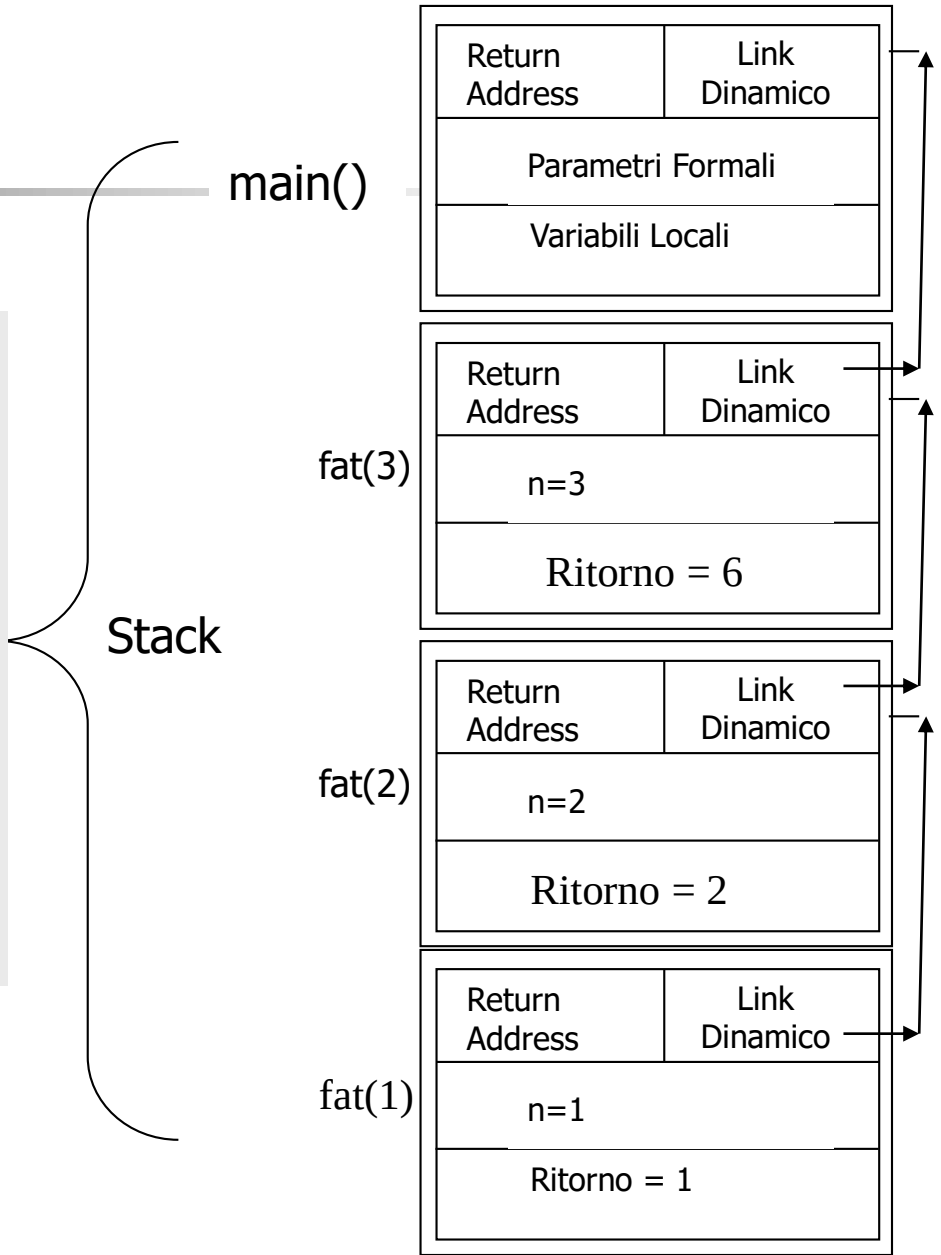
```
#include <stdio.h>

unsigned long numero;

double fat(unsigned long n){
  if (n>1) return(n*fat(n-1));
  else return (1);
}

int main(void)
{
  printf("\nCALCOLO DI n! ");
  printf("\nInserisci un numero: ");
  scanf("%u", &numero);
  printf("\nFattoriale di %d e' %e", numero, fat(numero));
}
```

Mentre la soluzione iterativa sarebbe stata:

```
fat=1.0;
while (n>1) fat*=n--;
```

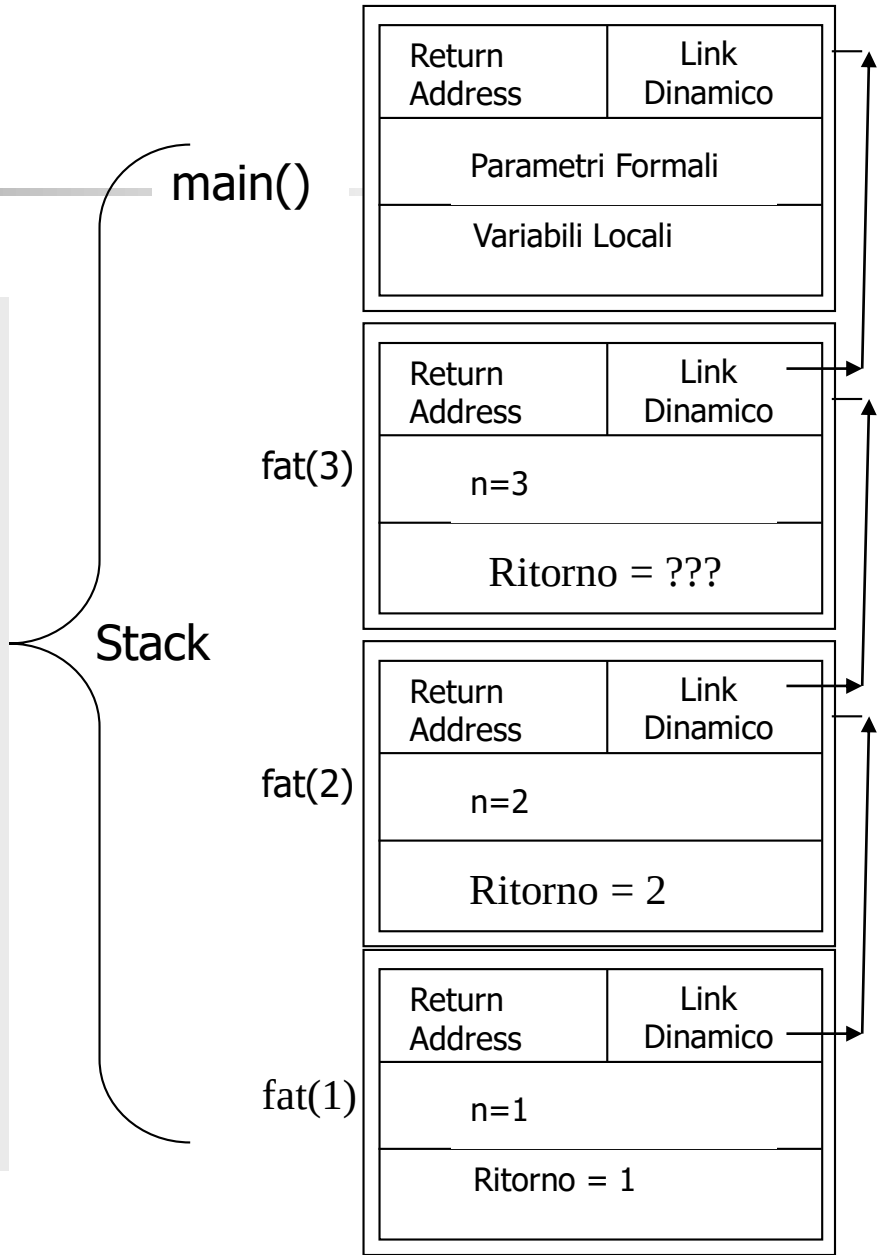
Ricorsione: Esecuzione

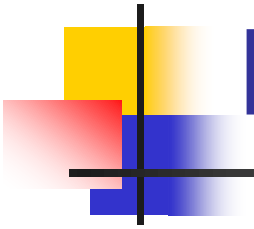
```
double fat(unsigned long n){
  if (n>1) return(n*fat(n-1));
  else return (1);
}

int main(void)
{
   printf("\n%e", fat(3));
}
```



Ricorsione: Esecuzione

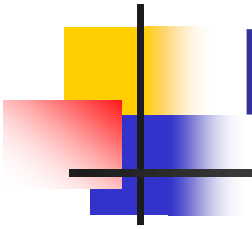
```
/*E se mi dimentico il return ?*/

double fat(unsigned long n){
  if (n>1) n*fat(n-1);
  else return (1);
}

int main(void)
{
   printf("\n%e", fat(3));
}
```





Ricorsione

- Ricerca Binaria
- Vettore ordinato (inf,…,sup)
- Si confronta l'elemento da cercare con l'elemento di indice med=(inf+sup)/2
- Se è quello, fine
- Se è <, si riapplica la ricorsione al vettore compreso tra gli indici inf e sup=med-1
- Se è >, si riapplica la ricorsione al vettore compreso tra gli indici inf=med+1 e sup
- Se inf>sup, allora l'elemento cercato non esiste, fine



Ricerca Binaria

```
unsigned short RicercaBinaria(float v[], int inf, int sup, float x){
  int medio=(inf+sup)/2;

  if (inf>sup) return 0;
  if (v[medio]==x) return 1;
  if (v[medio]>x)  return RicercaBinaria(v,inf,medio-1,x);
  return RicercaBinaria(v,medio+1,sup,x);
}
```