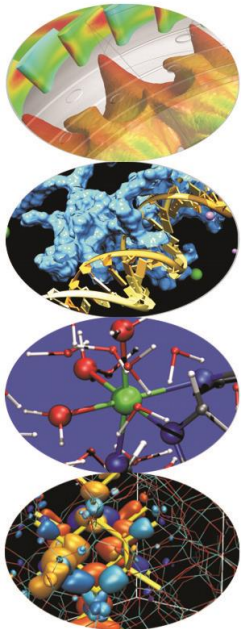
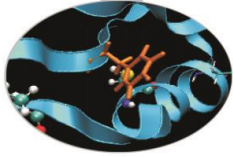




# Approfondimenti sulle procedure

esercitazioni

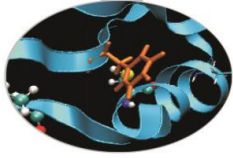




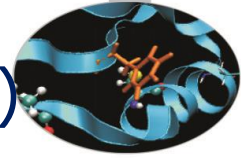
# Procedure interne - esercizi

- Scrivere un programma che calcola la somma elemento per elemento di 2 vettori; l'inizializzazione degli operandi e la stampa del risultato venga realizzata con subroutine interne.
- Scrivere un programma che, partendo da un vettore sufficientemente lungo, somma:
  - tutti gli elementi
  - gli elementi di posto dispari
  - gli elementi di posto pari
  - Realizzare i 3 calcoli con funzioni interne distinte, di nome *SommaTuttiEll*, *SommaDispariEll*, *SommaPariEll*, che non richiedano l'utilizzo di argomenti.

# Procedure esterne e interfacce



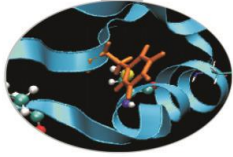
- Riprendendo l'esercizio precedente, riscrivere le funzioni *SommaTuttiEll*, *SommaDispariEll*, *SommaPariEll* come funzioni esterne, che ovviamente richiedono argomenti e un'interfaccia esplicita. Fare attenzione a dichiarare correttamente l'intento degli argomenti.



## Procedure esterne e interfacce (Fortran 2003)

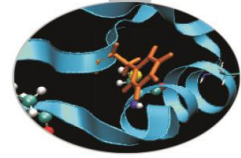
- Realizzare un MODULE *dati* contenente la definizione di un vettore (e della sua dimensione come costante PARAMETER).
- Realizzare in questo modulo le funzioni *inizializza*, *somma* e *stampa*.
- Realizzare quindi la funzione esterna *media*, che calcola il valor medio degli elementi. Attenzione: essendo una funzione esterna, nel modulo *dati* deve comparire solo la sua interfaccia. Suggerimento: fare uso dell'istruzione IMPORT.
- Realizzare quindi un PROGRAM per provare il codice scritto.

# Procedure elementali in Fortran 2003



Realizzare la funzione *scambia*, che scambia i valori tra i suoi 2 argomenti.

Dichiararla ELEMENTAL, quindi verificare che si può usare con argomenti sia scalari che vettoriali. Nota: se la funzione è esterna, è necessaria un'interfaccia esplicita.

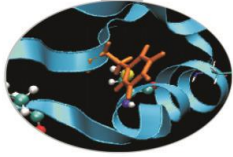


# Argomenti di tipo procedura

## Ordine degli argomenti

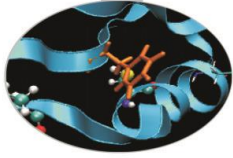
Realizzare la funzione *area*, che calcola l'area sottostante una funzione. Suggestimenti: la funzione da calcolare sia descritta con una *function*, da passare in argomento ad *area*; gli altri argomenti siano i valori inizio e fine intervallo d'integrazione.

Utilizzare la funzione *area* in un programma che la usi con gli argomenti in diverso ordine.



# Argomenti facoltativi

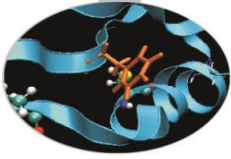
- Realizzare la funzione *area*, dell'esercizio precedente, in modo che tutti gli argomenti, tranne la funzione, siano facoltativi. Modificare il programma in modo che la richiami con e senza argomenti.
- Realizzare una procedura che stampi i multipli di un intero; il valore dell'intero e il numero dei multipli stampati siano facoltativi.



# Argomenti di tipo procedura

- Scrivere le funzioni *somma* e *prodotto* che sommano e moltiplicano 2 valori passati in argomento.
- Realizzare quindi la procedura *stampa* che accetta in argomento una funzione, 2 valori e ne stampi a video il risultato.
- Scrivere un programma che permetta di presentare il risultato di *somma* e *prodotto* utilizzando *stampa*.

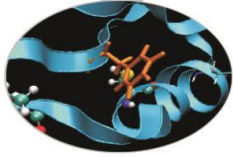




# Argomenti allocabili (Fortran 2003)

- Realizzare un programma che utilizzi le procedure:
  - Inizializza* - alloca la memoria per 2 vettori e li inizializza
  - calcola* – ritorna il prodotto scalare dei 2 vettori
  - libera* – rilascia la memoria allocata
- Richiamare le 3 procedure ripetutamente all'interno di un ciclo iterativo.
- Dichiarare i vettori come ALLOCATABLE nel programma e nelle procedure *inizializza* e *libera*.

# Funzioni vettoriali (Fortran 2003)

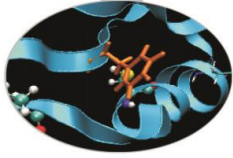


Realizzare la funzione esterna *calcola* che ritorna la matrice risultato del calcolo  $A + B \times C$ , dove A, B, C sono matrici e l'operazione "x" è la moltiplicazione matriciale righe per colonne.

Scrivere versioni differenti della funzione, che riservano la memoria per il risultato ritornato e eventuali risultati intermedi mediante: allocazione automatica, allocazione dinamica con oggetti POINTER o ALLOCATABLE.

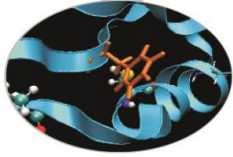
Usare queste versioni in un programma che confronta i risultati ottenuti.

Nota: poiché le funzioni sono esterne, occorre scriverne l'interfaccia esplicita.



# Funzioni ricorsive

Scrivere una funzione, realizzata come ricorsiva, che ritorni il termine N-esimo della successione di fibonacci, definita come: 1, 1, 2, 3, 5, ...  $N = [N-1] + [N-2]$

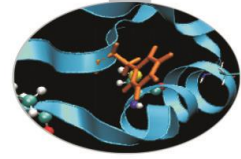


# Funzioni ricorsive

I frattali sono figure geometriche autosomiglianti, nel senso che, osservandole a scale diverse, si assomigliano.

Uno dei metodi più semplici per realizzare un frattale consiste nel considerare una matrice quadrata abbastanza grande, dividerla in 4 parti uguali e azzerare la parte in alto a destra (colonne con indice maggiore e righe con indice minore). Quindi, per ognuna delle 3 parti rimanenti, si itera il procedimento.

È possibile costruire questo frattale con una procedura ricorsiva.



# Funzioni ricorsive

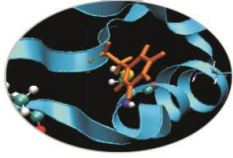
Suggerimenti:

In un modulo si dichiara una matrice quadrata con dimensione potenza del 2 (ad esempio 64). Nel programma si azzera la matrice.

Si scrive una procedura ricorsiva che opera su una matrice quadrata virtuale, con le stesse dimensioni di quella condivisa. Nella procedura la matrice viene suddivisa in 4 parti uguali, quindi la procedura richiama se stessa per ognuna delle 3 parti che non devono essere nulle.

Se invece le dimensioni sono diventate troppo piccole ( $\leq 2$ ), gli elementi corrispondenti della matrice del modulo vengono posti a 1.

Infine il programma presenti a video la matrice così ottenuta.

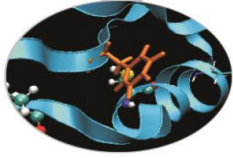


# Funzioni ricorsive

Malgrado gli algoritmi ricorsivi richiedano grandi risorse di calcolo, per la soluzione di problemi poco più che banali, sono tuttavia molto comodi per risolvere alcuni tipi di problemi, che altrimenti sarebbero molto più difficilmente risolvibili.

Un esempio abbastanza tipico è la cosiddetta Torre di Hanoi. Una descrizione del gioco e dell'algoritmo, per chi vuole cimentarsi nella realizzazione della versione Fortran, si può trovare alla pagina WEB:

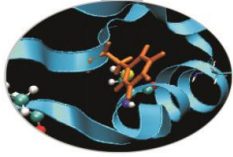
<http://www.cut-the-knot.org/recurrence/hanoi.shtml>



# Nomi generici

Realizzare 3 versioni (per matrici INTEGER, REAL(4), REAL(8)) di una funzione che ritorna la matrice risultato del calcolo  $A + B \times C$ , dove A, B, C sono matrici e l'operazione "x" è la moltiplicazione matriciale righe per colonne.

Associare il nome generico calcola alle 3 funzioni e scrivere un programma che richiami la funzione calcola con matrici di tipo diverso.

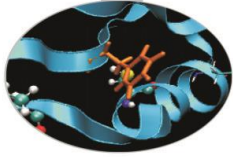


# Nomi generici (Fortran 2003)

Realizzare 3 versioni (INTEGER, REAL(8), COMPLEX(8)) di una funzione esterna che ritorna il valor medio degli elementi di un vettore.

Scrivere un programma che permette di usare queste funzioni con il nome generico *ValorMedio*.



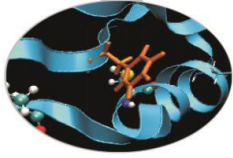


# Procedure intrinseche

Riprendendo il programma scritto per generare il quadrato frattale, aumentare le dimensioni della matrice fino a 512 o 1024, quindi usare la funzione intrinseca `SYSTEM_CLOCK` per misurare il tempo impiegato per generare la figura.

Attenzione: conviene evitare la stampa finale o ridimensionarla opportunamente.

Come ulteriore esercizio sostituire o affiancare `SYSTEM_CLOCK` con `CPU_TIME`.



# Procedure intrinseche

- Realizzare un programma che, in un ciclo iterativo indefinito, chieda di indovinare un numero da 1 a 10, da confrontare quindi con il valore ritornato da `RANDOM_NUMBER` (opportunamente rinormalizzato).
- Nel programma precedente usare `RANDOM_SEED` per modificare il seme con il valore dei millisecondi ritornato da `DATE_AND_TIME`. Può essere simpatico usare preventivamente `DATE_AND_TIME` per presentare data e ora di esecuzione del programma.