

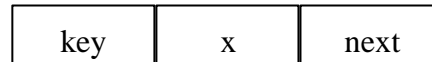
CdL in Scienze dell'Informazione, sede di Cesena
Sistemi per l'Elaborazione dell'Informazione 1
Prof. M. Roccetti

11 luglio 2000

ESERCIZIO NUMERO 4

Data una struttura a lista, implementata mediante la struttura dati NODO seguente:

```
typedef struct S_NODO {  
    int      key;  
    double   x;  
    S_NODO  *next;  
} NODO;
```

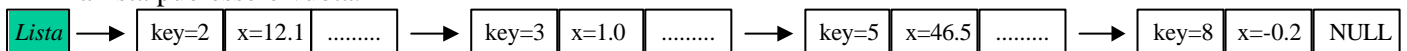


Sia *Lista* la radice della lista (NODO *Lista), *Lista* assume valore NULL quando la lista è vuota, ogni nodo della lista può avere un nodo successivo puntato dal puntatore next, che assume valore NULL se il nodo successivo corrispondente non esiste.

I nodi della lista sono ordinati secondo la chiave key in modo crescente.

Non esistono due nodi che abbiano la stessa chiave key.

La lista può essere vuota.



Implementare la funzione

```
int trova( int KEY, double *pX, NODO *pnodo );
```

che verrà chiamata ad. es. nel modo seguente:

```
NODO *Lista=NULL;
```

```
int      ris, KEY;
```

```
double   X;
```

```
.....
```

```
..... costruzione della lista .....
```

```
.....
```

```
KEY = 5;
```

```
ris = trova ( KEY, &X, Lista);
```

La funzione *trova* scorre la lista che discende dal parametro pnodo, alla ricerca di un nodo che abbia chiave key uguale a KEY.

Se tale nodo esiste, la funzione scrive il valore x del nodo nella locazione di memoria puntata da pX, e restituisce 1.

Se tale nodo non esiste, la funzione restituisce 0.

OSS. Il puntatore *pnodo* passato come terzo argomento alla funzione può valere NULL.

OSS. nell'esempio descritto la funzione restituisce 1 e assegna il valore 46.5 ad X.

SOLUZIONE dell'ESERCIZIO 4

```
int trova( int KEY, double *pX, NODO *pnodo )
{
    if(pnodo)
    {
        if( pnodo->key==KEY ) {
            *pX=pnodo->x;
            return(1);
        }
        else {
            if( pnodo->key < KEY )
                return ( trova(KEY,pX,pnodo->next) );
            else
                return(0);
        }
    }
    else
        return(0);
}
```