

Printed: Mercoledì, 20 maggio 2015 18:58:07

```
/*
Sequenza (ad allocazione dinamica) [di interi] con inserimento in coda e
doubly linked per scorrimento dalla coda e dalla testa.
Sequenza MODIFICABILE sia nei componenti sia nella struttura.
Vedi anche seq.h (di cui mancano le definizioni delle operazioni: size, at,
append, qui inutili)
*/

#include <stdio.h>
#include <stdlib.h>

struct MutSElem {
    struct MutSElem * pred;
    struct MutSElem * next;
    int val;
};

typedef struct MutSElem elem;
typedef elem * intMutS;

/*
struct CoppiaIntMutS {
    intMutS left;
    intMutS right;
};

typedef struct CoppiaIntMutS iPair;
typedef iPair * intMutSPair;
*/

/* ----- OPERAZIONI ----- */

/* ----- COSTRUZIONE DELLA STRUTTURA ----- */

intMutS empty(){
    /* crea sequenza/elemento vuota/o */
    return NULL;
};

intMutS addTail(intMutS u, int x){
    /* aggiunge in coda a elemento u */
    intMutS r;
    r = (intMutS)malloc(sizeof(elem));
    r->pred = u;
    r->val = x;
    if (isEmpty(u)) r->next = u;
    else {
        r->next = u->next;
        u->next = r;
    }
    return r;
};

/* ----- OSSERVAZIONE COMPONENTI ----- */

int isEmpty(intMutS u){
    /* sull'elemento o sulla sequenza */
    return (u == NULL);
};
```

Printed: Mercoledì, 20 maggio 2015 18:58:07

```
intMutS moveR(intMutS u){
    /* restituisce elemento a destra dell'argomento, se esiste */
    if (isEmpty(u)) return u;
    return u->next;
};

intMutS moveL(intMutS u){
    /* restituisce elemento a sinistra dell'argomento, se esiste */
    if (isEmpty(u)) return u;
    return u->pred;
};

int val(intMutS u){
    /* valore dell'elemento u */
    if (isEmpty(u)) return 0;
    return u->val;
};

/* ----- MODIFICA COMPONENTI ----- */

void reset(intMutS u, int v){//pubblica
    /* reset a v del valore dell'argomento, se esiste */
    if (isEmpty(u)) return;
    u->val = v;
    return;
};

void swap(intMutS u, intMutS w){//pubblica
    /* scambio dei valori associati ai due argomenti, se esistono */
    if (isEmpty(u)||isEmpty(w)) return;
    int temp;
    temp = u->val;
    u->val = w->val;
    w->val = temp;
    return;
};

/* ----- I/O ----- */

intMutS readIntSeq(){//pubblica
    /* Legge sequenza di coppie: intero, seguito da ','. La sequenza è
       delimitata a sinistro da '[', e a destra da un qualunque carattere,
       preferibilmente ']'
       */
    intMutS inf = empty();
    intMutS sup = empty();
    char c;
    if ((scanf("%c",&c)!=1)|| (c!='[')){
        printf("Carattere inatteso: la sequenza deve avere la forma: [xx,...,xxc\n");
        return sup;
    }
    /* primo elemento */
    int n;
    int endTest = 1; // 1 per true
    while (endTest && (scanf("%d",&n)==1)){
        sup = addTail(sup,n);
        if (isEmpty(inf)) inf = sup;
        endTest = scanf("%c",&c)==1 && (c==',' );
    }
}
```

Printed: Mercoledì, 20 maggio 2015 18:58:07

```
        if (endTest) scanf("%c",&c);
        return sup;
    }
}

void innerWrite(intMutS inf, intMutS sup){//privata
// privata di writeIntSeq
/* l'argomento è l'estremo inferiore */
if (inf == sup) return;
printf("%d,",val(inf));
innerWrite(moveR(inf),sup);
return;
};

void writeIntSeq(intMutS inf, intMutS sup){//pubblica
if (isEmpty(sup)){
    printf("[]\n");
    return;
}
printf("[");
innerWrite(inf,sup);
printf("%d]\n",val(sup));
return;
};
```