

Printed: Lunedì, 13 aprile 2015 18:18:07

```

/*
Sequenza (ad allocazione dinamica) [di interi] con inserimento in coda.
Istanziamento, per il C, della sequenza generica (i.e. di valori di generico tipo t).
*/

/*
LIMITI (della struttura).

(1) Non e' un ADT. Infatti la rappresentazione è completamente esposta:
    int fakeSum(intSeq u){
        u->body = Null;
        return tail(u);
    }
questa definizione inserita in un qualunque punto del programma esterno al nostro "ADT"
e' compilata, e quando eseguita permette di accedere la rappresentazione e modificarla.

(2) Non e` un MODULO. Infatti, nessuna distinzione tra:
- cio che e' visibile ed esportabile e cio' che e' privato (ovvero introdotto
ad uso esclusivo delle implementazione data per Seq;
Esempi: Tutte le definizioni con prefisso Inner sono private.
- Le definizioni non sono Fully Abstract.
- Manca di LOCALIZZAZIONE. La definizione innerAt e' pensata per essere
usata in esclusiva da At, in particolare e' stata progettata per essere
invocata nelle condizioni di stato e sullo spazio di valori previsto
dal suo impiego in At.
Esempio. InnerWrite stampa "[" quando applicato alla sequenza vuota non "[]"
- STRUTTURA PIATTA che non ci aiuta distinguere le definizioni principali dalle
altre
- Struttura piatta che rende difficile intervenire per fare testing, modifiche,
manutenzione in generale.
Esempio. Volendo modificare At mi devo ricordare che innerAt e` un suo componente
e va modificato anch'esso o rimosso.
- Struttura piatta che richiede commenti aggiuntivi solo per indicare quale forma
dovrebbe avere la struttura.
*/

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

//volendo usare il tipo bool (in headE, readIntSeq) invece di int
//enum bool {false, true};
//typedef enum bool bool;

//Le annotazioni "privata", "privata di..", "pubblica" sono solo commento e non
//hanno alcun effetto sul codice prodotto: C non ha questi livelli di visibilita'

struct seqElem { //privata
    struct seqElem * body;
    int tailVal;
};

typedef struct seqElem elem; //privata
typedef elem * intSeq; //privata

/* ----- OPERAZIONI ----- */
/* ----- COSTRUTTORI: COSTRUZIONE DELLA STRUTTURA ----- */

```

Printed: Lunedì, 13 aprile 2015 18:18:07

```
intSeq empty(){//pubblica
    return NULL;
};

int isEmpty(intSeq u){//pubblica
    return (u == NULL);
};

intSeq add(intSeq u, int x){//pubblica
    intSeq r;
    r = (intSeq)malloc(sizeof(elem));
    r->body = u;
    r->tailVal = x;
    return r;
};

/* ----- PRODUTTORI: OSSERVAZIONE COMPONENTI ----- */

intSeq head(intSeq u){//pubblica
    if (isEmpty(u)) return 0;
    return u->body;
};

/*
bool headE(intSeq *u){
    Equivalente ad head sopra ma con trattamento delle
    eccezioni in stile C: vedi uso SeqTest.c
    if (isEmpty(*u)) return false;
    *u = (*u)->body;
    return true;
*/

int tail(intSeq u){//pubblica
    if ((isEmpty(u)) return 0;
    return u->tailVal;
};

int size(intSeq u){//pubblica
    if (isEmpty(u)) return 0;
    return 1+size(head(u));
};

int at(intSeq u, int n){//pubblica
    /* indefinita se n<0 oppure n>size(u) */
    /* qui definizione di innerAt */
    if (n>=0 & n<=size(u))
        return innerAt(u,size(u)-n);
};

int innerAt(intSeq u, int k){//privata
    // privata di at
    if (k==0) return tail(u);
    return innerAt(head(u),k-1);
};

/* ----- MODIFICATORI: MODIFICA COMPONENTI ----- */

intSeq append(intSeq u, intSeq w){//pubblica
    if (isEmpty(u)) return w;
    else if (isEmpty(w)) return u;
```

```
    return add(append(u,head(w)),tail(w));
};

/* ----- PRODUTTORI: I/O ----- */

intSeq readIntSeq(){//pubblica
    /* ausiliaria per I/O */
    /* sequenza di coppie intero ',' con delimitatore sinistro '[',
       e delimitatore destro, un qualunque carattere bv */
    intSeq r = empty();
    char c;
    if ((scanf("%c",&c)!=1)|| (c!='[')){
        printf("Carattere inatteso: la sequenza deve avere la forma: [xx,...,xxc\n");
        return r;
    }
    /* primo elemento */
    int n;
    int endTest = 1; // 1 per true
    while (endTest && (scanf("%d",&n)==1)){
        r = add(r,n);
        endTest = scanf("%c",&c)==1 && (c==',' );
    }
    if (endTest) scanf("%c",&c);
    return r;
}

void innerWrite(intSeq u){//privata
    // privata di writeIntSeq
    if (isEmpty(u)){
        printf("[");
        return;
    }
    innerWrite(head(u));
    printf("%d,",tail(u));
    return;
};

void writeIntSeq(intSeq u){//pubblica
    /* ausiliaria per I/O */
    /* qui definizione di innerWrite */
    if (size(u)==0){
        printf("]\n");
        return;
    }
    innerWrite(head(u));
    printf("%d]\n",tail(u));
    return;
};
```