

Code



Code

- Una coda è un insieme ordinato, non limitato, di elementi dello stesso tipo dove il primo elemento memorizzato e' anche il primo ad essere recuperato (FIFO=first input first output)



ADT coda

■ Il tipo CODA è un ADT $\langle S, F, C \rangle$ dove

$S = \{\text{coda}, \text{atomo}, \text{boolean}\}$

- coda è il dominio di interesse, atomo è il dominio degli elementi che formano le liste

$F = \{\text{incoda}, \text{front}, \text{null}, \text{outcoda}\}$

- incoda : atomo x coda \rightarrow coda
 - inserisce un elemento alla fine della coda
- front : coda \rightarrow atomo
 - ritorna l'elemento in cima alla lista
- null : coda \rightarrow boolean
 - ritorna il valore vero se la coda è vuota
- outcoda : coda \rightarrow coda
 - ritorna la coda privata dell'elemento in cima

$C = \text{coda vuota},$

- è la costante che denota la coda priva di elementi
-
-

Implementazione concatenata

```
typedef int TAtomo;
typedef struct Stelem {
    struct Selem *next;
    TAtomo info;
} elem;
typedef struct StCoda {
    elem *primo, *ultimo;
} TCoda, *PTCoda;

int null( TCoda C ) { return !C.primo;}
int outcoda ( PTCoda PC ) {
    elem *aux;
    if( null(*PC) ) return 0;
    aux = PC ->primo
    PC -> primo = PC ->primo ->next
    if (null(*PC)) PC ->ultimo = NULL
    free(aux);
    return 1;
}
```

```
TAtomo front( TCoda C ) {  
    if( null(C) ) return NULL;  
    return C.primo -> info;  
}
```

```
int incoda( PTCoda PC, TAtomo A ) {  
    elem *aux;  
    aux = (elem *)malloc(sizeof(elem));  
    if (!aux) return 0 ;  
    aux ->info = A;  
    aux -> next = NULL;  
    if null(*PC) PC ->ultimo = PC->primo = aux;  
    else { PC ->ultimo -> next = aux;  
          PC ->ultimo = aux  
          }  
    return 1;  
}
```
