

LISTA COLLEGATA

```

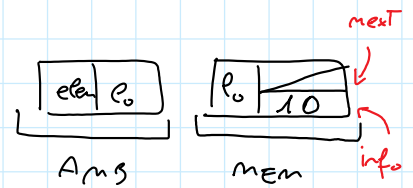
struct elemento { int info; struct elemento* next; }
typedef struct elemento ElementoDiLista;
typedef ElementoDiLista* ListaDiElementi;

```

```

main() {
    ElementoDiLista elem;
    elem.info = 10;
    elem.next = NULL;
}

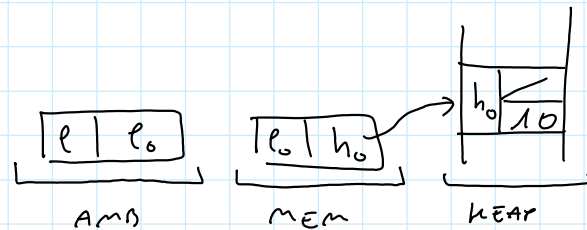
```



```

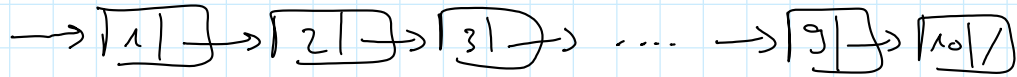
main() {
    ListaDiElementi l = malloc(sizeof(ElementoDiLista));
    l->info = 10;
    l->next = NULL;
}

```



... -> ... ≡ (*...). ...

SCRIVIAMO UN PROGRAMMA CREA UNA LISTA DI 10 ELEMENTI CONTENENTI I NUMERI DA 1 A 10



```
struct elemento { int info; struct elemento * next; }
```

```
typedef struct elemento ElementoDiLista;
```

```
typedef ElementoDiLista * ListaDiElementi;
```

```
main() {
```

```
ListaDiElementi lista = NULL;
```

```
ListaDiElementi new =
```

```
malloc(sizeof(ElementoDiLista));
```

```
new->info = 1;
```

```
lista = new;
```

```
for (int i = 2; i <= 10; i++) {
```

```
new->next = malloc(sizeof(ElementoDiLista));
```

```
new = new->next;
```

```
new->info = i;
```

```
}
```

Ci vogliono sempre e vanno scritte fuori dalle funzioni

inizialmente la lista è vuota

inserisce il 1° elemento nella lista

(devo modificare il collegamento al 1° elemento)

creo il 1° elemento usando il puntatore "temporaneo" new

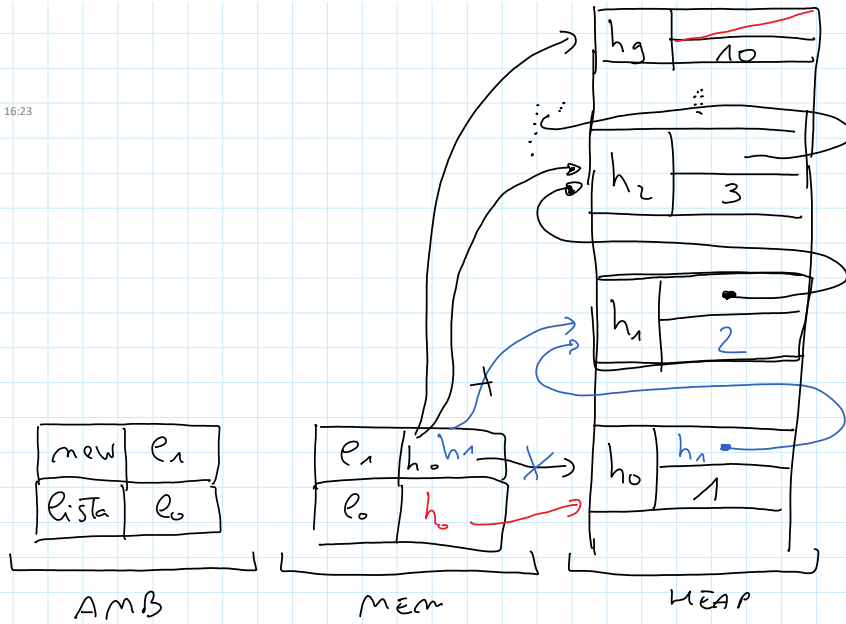
che pointer sempre all'ultimo elemento inserito

nel campo next dell'ultimo elemento inserito scrivo il puntatore a un nuovo elemento

new → next = NULL;

}

IL PRIMO ELEMENTO L'HO CREATO
FUORI DAL CICLO PER POTER
AGGIORNARE LISTA

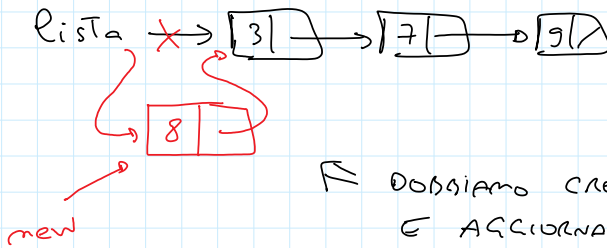


DEVO SCRIVERE NULL NEL CAMPO NEXT DELL'ULTIMO ELEMENTO

PUNTIATORE ALL'INIZIO DELLA LISTA (PRIMO ELEMENTO)

- GLI ELEMENTI SONO NELLO HEAP
- OGNI ELEMENTO HA UN PUNTIATORE AL SUCCESSIVO

AGGIUNGERE UN ELEMENTO "IN TESTA" AD UNA LISTA (in prima posizione)



```

    Chiamata
    addT(lista, 8)
  
```

⇒ DOBBIAMO CREARE L'ELEMENTO
E AGGIORNARE I PUNTORI

VALORE
DA
AGGIUNGERE

```

    lista Di Elementi new = malloc(sizeof(Elemento Di lista));
  
```

```

    new->info = x;
  
```

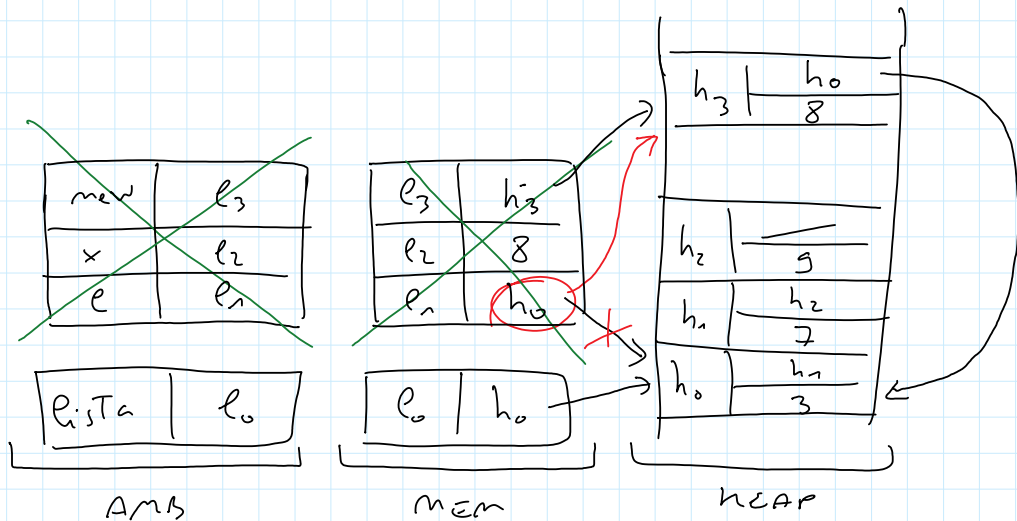
```

    new->next = e;
  
```

```

    e = new;
  
```

}



PROBLEMA $e = new;$ modifica l_0

copia di lista, non lista stessa

PER POTER MODIFICARE LISTA DEVO

PASSARLA ALLA PROCEDURA PER INDIRIZZO!!

```
void addT (ListaDiElementi *l, int x) {
```

```
    ListaDiElementi new = malloc (sizeof (ElementoDiLista));
```

```
    new -> info = x;
```

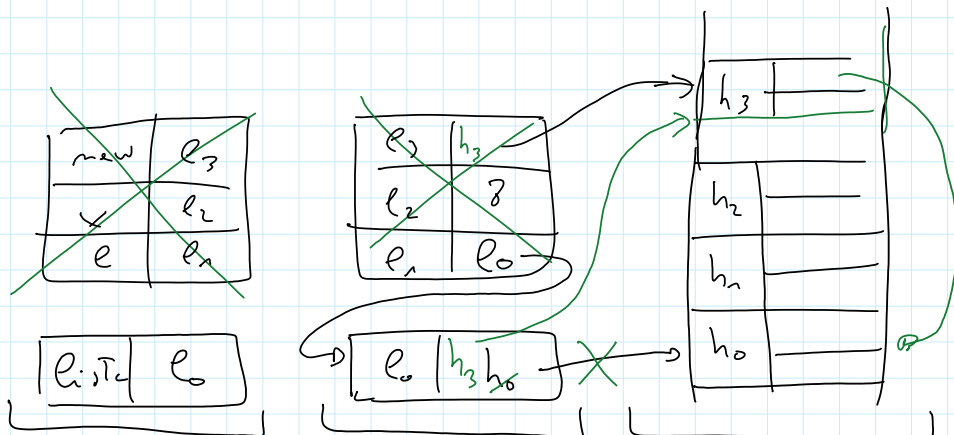
```
    new -> next = *l;
```

```
    *l = new;
```

```
}
```

CHIAMATA

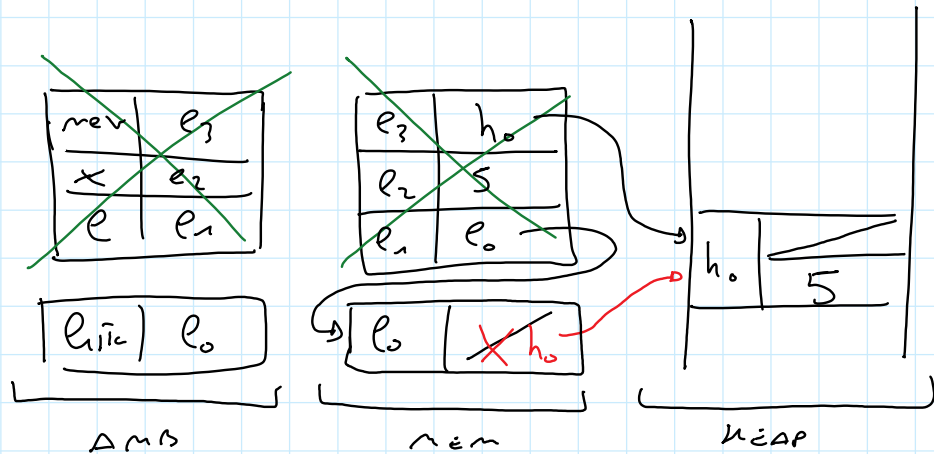
addT (&lista, 8)



E SE LA LISTA È VUOTA?

(Una lista vuota è rappresentata da un puntatore NULL)

```
main() {  
    ListaDiElementi lista = NULL;  
    addT ( &lista , 5 );  
    ...  
}
```



TUTTO OK
FUNZIONA ANCHE
CON LISTA VUOTA

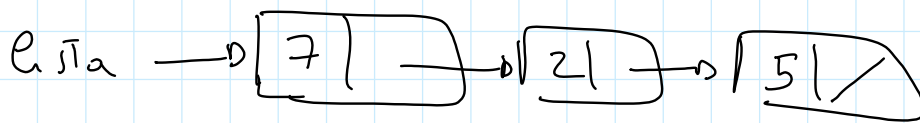
POSSIAMO USARE ADDT PER COSTRUIRE UNA LISTA

```

main ( ) {
    ListaDiElementi lista = NULL ;
    addT (&lista, 5);
    addT (&lista, 2);
    addT (&lista, 7);
}

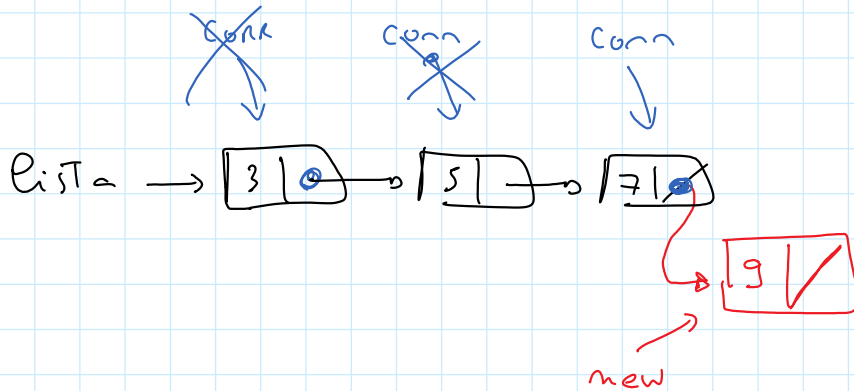
```

COSTRUISCE



GLI ELEMENTI SARANNO
DISPOSTI IN ORDINE
INVERSO RISPETTO A
COME LI HO INSERITI
(INSERISCO IN TESTA)

AGGIUNGERE UN ELEMENTO "IN CODA" AD UNA LISTA (IN ULTIMA POSIZIONE)



```
void addC (ListaDiElementi *l, int x)
```

PASSO PER INDIRIZZO PERCHÉ NEL CASO LA LISTA SIA VUOTA DEVO MODIFICARE LA VARIABILE LISTA

```
ListaDiElementi new =
    malloc(sizeof(ElementoDiLista));
```

```
new->info = x;
new->next = NULL;
```

```
if (*l == NULL)
    *l = new;
```

CASO LISTA VUOTA

```
else {
    ListaDiElementi curr = *l;
    while (curr->next != NULL) {
        curr = curr->next;
    }
```

PUNTIATORE TEMPORANEO CHE USO PER SCORRERE LA LISTA (INIZIALMENTE PUNTA AL PRIMO ELEMENTO)

FINITANTO CHE CURR NON PUNTA ALL'ULTIMO ELEMENTO (QUELLO CHE HA NEXT = NULL)

```
curr->next = new;
```

ALLA FINE DEL CICLO CURR PUNTA ALL'ULTIMO ELEMENTO

ATTENZIONE: $conn \rightarrow next$ nella guardia
del while non da errore
perché sono sicuro che

$conn \neq NULL$

SE $conn$ POTESSE ESSERE
NULL L'OPERAZIONE

$conn \rightarrow next$ ($conn \rightarrow info$)

POTREBBE DARE ERRORE A
TEMPO DI ESECUZIONE!!

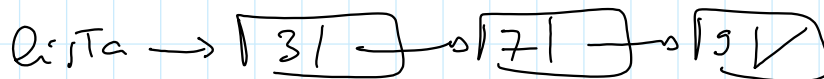
$(*conn).next$

$*NULL$ È ERRORE!!

A QUESTO PUNTO POSSO USARE ADDC PER COSTRUIRE UNA LISTA

```
main() {
    listaDiElementi lista = NULL;
    addC(&lista, 3);
    addC(&lista, 7);
    addC(&lista, 9);
    :
}
```

COSTRUISCE



GLI ELEMENTI VENGONO INSERITI NELL'ORDINE IN CUI LI HO PASSATI AD ADDC.

ATTENZIONE: COSTRUIRE UNA LISTA AGGIUNGENDO GLI ELEMENTI IN CODA È MENO EFFICIENTE!!
OGNI CHIAMATA DI ADDC DEVE SCANDIRE LA LISTA DALL'INIZIO.