

Fondamenti di Informatica e Laboratorio T-AB
Ingegneria dell'Automazione
a.a. 2008/2009

Lab 09

Puntatori

Esercizio 1

1. Si legga un numero intero n da input.
2. Si definisca un vettore V di n puntatori a char.
3. Si leggano n caratteri da input e si salvi ciascun carattere in una variabile dinamica creata appositamente, il cui indirizzo è conservato in un elemento di V .
4. Si stampino i caratteri puntati dagli elementi di V di indice dispari, seguiti dai caratteri puntati dagli elementi di V di indice pari.
5. Si liberi la memoria dalle strutture dati create.

Esercizio 2

Si scriva una funzione che:

1. accetti come unico parametro un puntatore a L (vedi caso visto a lezione il cui codice è riportato nella slide successiva),
2. restituisca in uscita il numero di strutture collegate l'una all'altra, considerando il valore NULL come indicatore del raggiungimento dell'ultimo elemento (nel caso visto a lezione il risultato sarebbe 10).

Esercizio Lezione

1. Si definisca un nuovo tipo di dato L, come record formato da un intero (campo value) e da un puntatore a L (campo next)
2. Si definisca una variabile p di tipo puntatore a L.
3. Mediante un ciclo, si definiscano 10 variabili dinamiche di tipo L, in modo che p punti alla 1^a variabile creata, il puntatore di tale variabile punti alla 2^a, il puntatore della 2^a punti alla 3^a variabile, etc. Si inizializzi l'ultimo puntatore al valore NULL.
4. Si inizializzino i campi value dei record creati alle potenze di due (21, 22, . . . , 210), nell'ordine dal primo all'ultimo.
5. Si mostrino tutti i valori inizializzati, dal primo all'ultimo.
6. Si liberi la memoria dalle strutture dati create.

Soluzione

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

```
// 1. definisci il tipo di dato
```

```
typedef struct L {
    int value;
    struct L *next;
} L;
```

```
// 3. 4. crea e inizializza n istanze di L
```

```
L* crea_lista( int n, int start_value ) {
    L *p=NULL, *last=NULL;
    int i;
    p=last=( L* )malloc( sizeof( L ) );
    p->value=start_value;
    p->next=NULL;
    for( i=1; i<n; i++ ) {
        last->next=( L* )malloc( sizeof( L ) );
        last->next->value=2*last->value;
        last->next->next=NULL;
        last=last->next;
    }
    return p;
}
```

// 5. mostra il contenuto

```
void mostra_contenuto( L *p ) {  
    while( p!=NULL ) {  
        printf( "%d\n", p->value );  
        p=p->next; }  
}
```

// 6. libera la memoria

```
void libera_memoria( L *p ) {  
    L *current=p;  
    while( current!=NULL ) {  
        p=current->next;  
        free( current );  
        current=p; }  
}
```

```
int main() {
```

// 2. definisci la variabile p

```
    L *p=NULL;  
    p=crea_lista( 10, 2 );  
    mostra_contenuto( p );  
    libera_memoria( p );  
}
```

Esercizio 3

1. Si definisca un nuovo tipo di dato LL, come record formato da un intero (campo ***value***) e da due puntatori a L (campi ***prev*** e ***next***).
2. Si definiscano 10 variabili dinamiche di tipo LL, in modo che ciascun elemento punti al precedente (con ***prev***) e al successivo (con ***next***). Si inizializzino il primo puntatore ***prev*** e l'ultimo ***next*** al valore NULL.
3. Si mostrino tutti i valori, percorrendo gli elementi prima in un senso poi nell'altro.
4. Si liberi la memoria dalle strutture dati create.