

CORSO DI PROGRAMMAZIONE
A.A. 2011-12

Dispensa 21

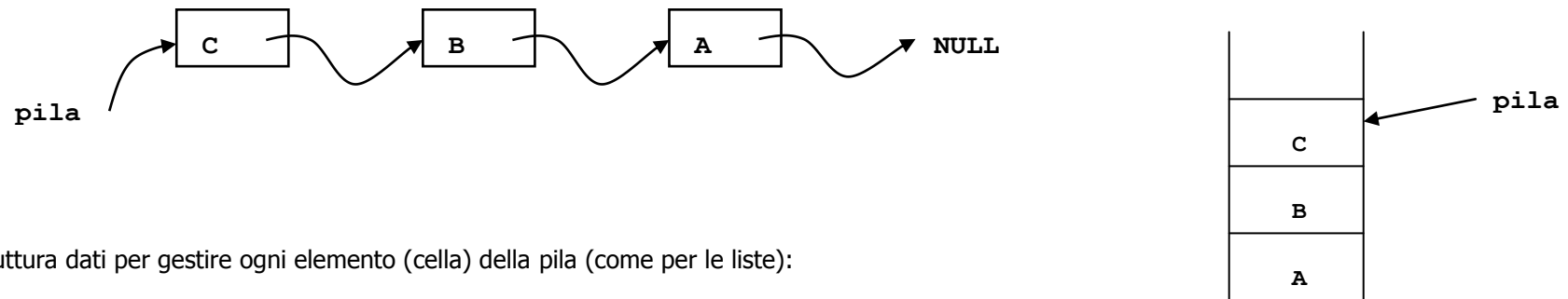
Laboratorio

Dott. Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it

<http://www.programmazione.info>

21.1 Le Pile

Una coda in sintesi è una lista (come descritta nella dispensa 17) quindi con un puntatore alla testa dell'elenco delle celle, con l'obbligo di eseguire tutte le operazioni a partire dalla testa e in particolare per ogni lettura l'elemento letto in testa viene eliminato dalla pila:



La struttura dati per gestire ogni elemento (cella) della pila (come per le liste):

```
struct cella{
    char valore;
    struct cella *next;
};

struct cella *pila = NULL; //puntatore alla testa
```

In una Pila tutti i nuovi elementi vengono inseriti in testa, tutte le altre operazioni vengono fatte sempre a partire dalla testa, ma ogni volta che si va a considerare un elemento questo viene tolto dalla pila.

Solitamente per operare con le pile si utilizzano due funzioni chiamate "push()" e "pop()", rispettivamente per l'inserimento e la lettura all'interno di una pila. Attraverso queste due funzioni si eseguiranno tutte le operazioni sulla pila. La pila è una struttura dati di tipo LIFO (Last In First Out) dove l'ultimo elemento ad essere inserito è il primo ad essere gestito (letto).

21.1.1 Funzione push()

La funzione push() serve per inserire un elemento all'interno della pila:

```
void push(struct cella **p, struct cella *nuovo);
```

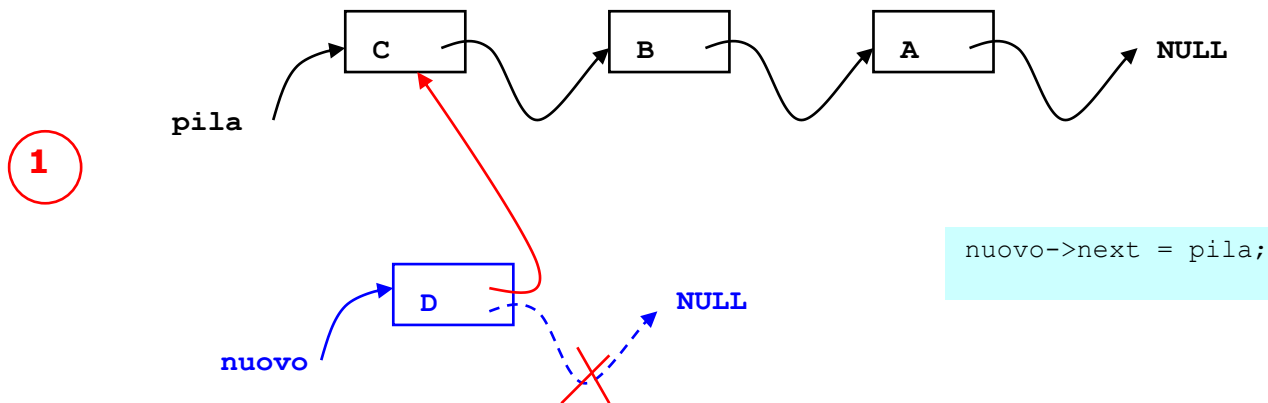
dove "p" è il riferimento alla pila da gestire e "nuovo" è il puntatore al nuovo elemento da inserire

La funzione push():

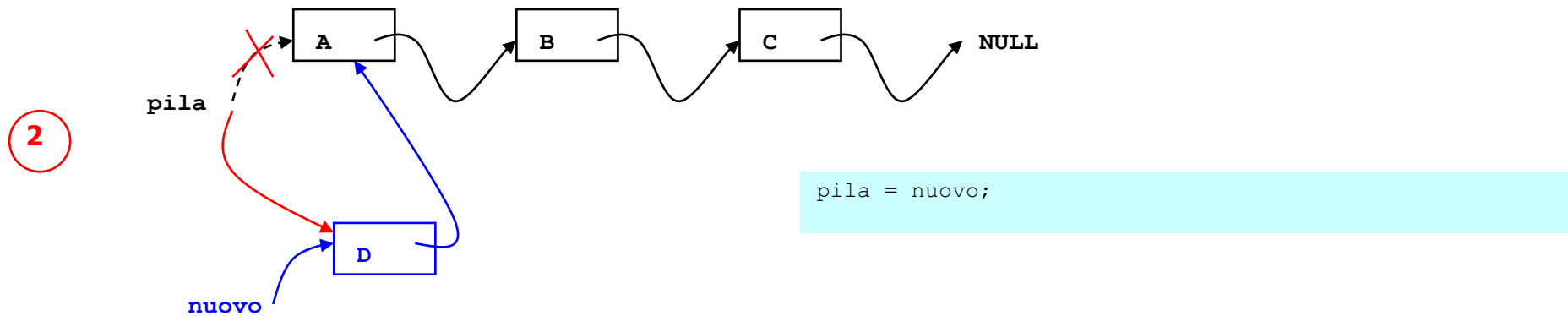
```
void push(struct cella **p, struct cella *nuovo)
{
    if (nuovo == NULL)
        return; //errore! non esiste un elemento da inserire!
    nuovo->next = *p;
    *p = nuovo;
}
```

Da notare che la funzione push() corrisponde ad un semplice inserimento in testa ad una lista visto nella dispensa 17.

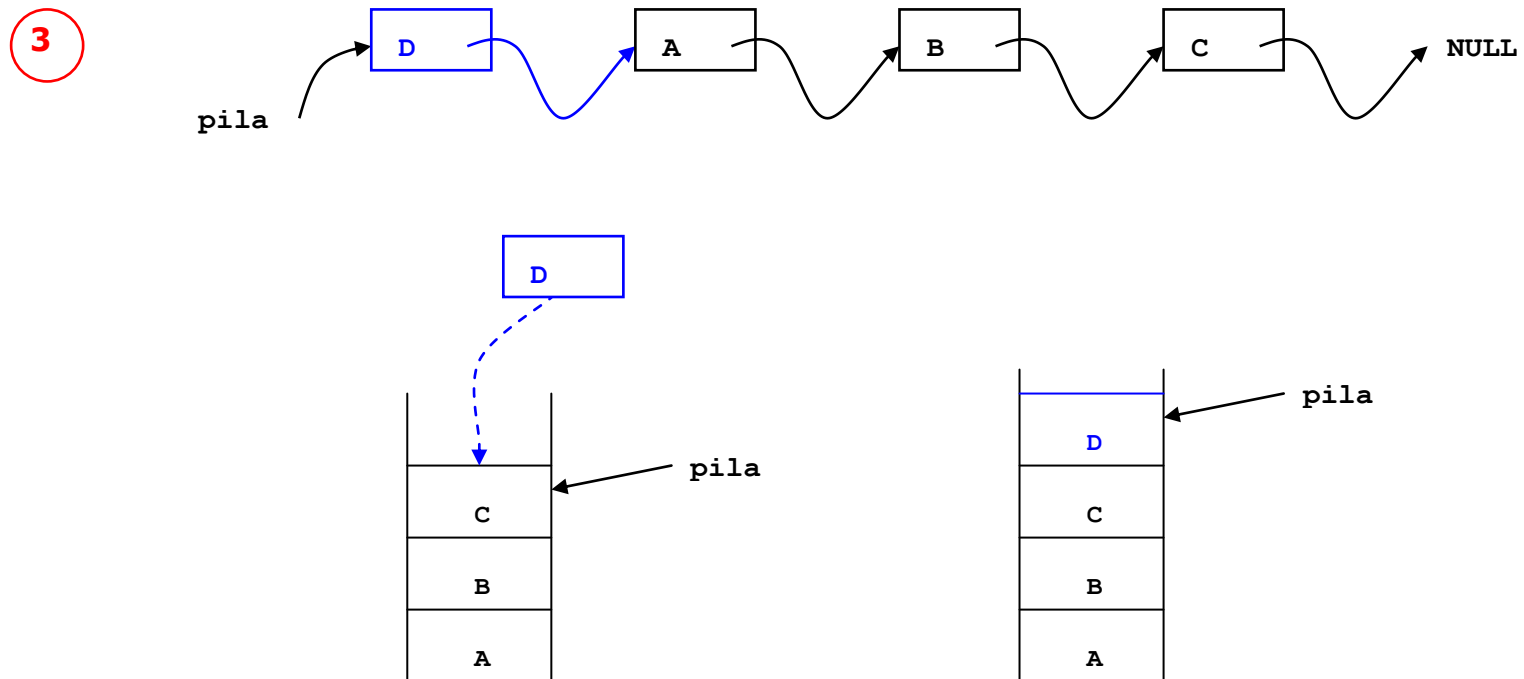
La prima operazione da compiere è collegare la nuova cella alla prima della lista, quindi fare in modo che l'elemento "next" della nuova cella prenda lo stesso valore di "pila":



La seconda operazione consiste nell'associare la testa al nuovo elemento creato:



In modo da ottenere l'inserimento del nuovo elemento in testa:



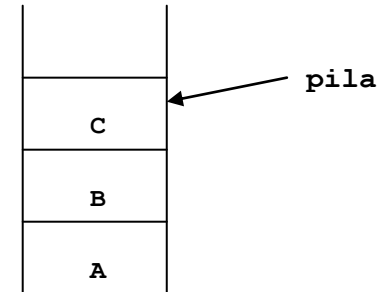
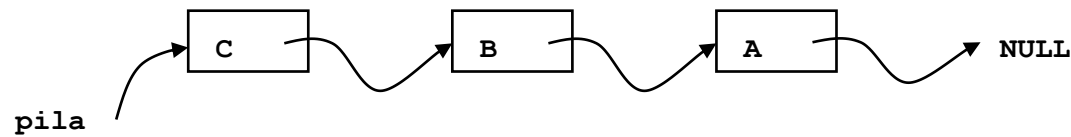
21.1.2 Funzione pop()

La funzione pop() consente di leggere il primo elemento in testa alla pila restituendo un puntatore, tale elemento viene eliminato dalla:

```
struct cella *pop(struct cella **p);
```

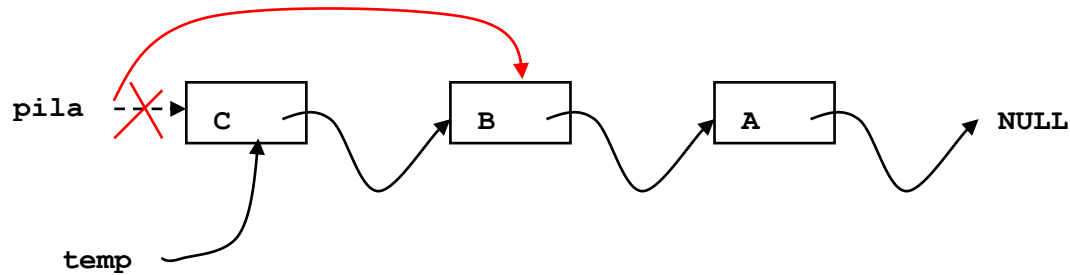
dove "p" è il riferimento alla pila da gestire.

Consideriamo la seguente pila:

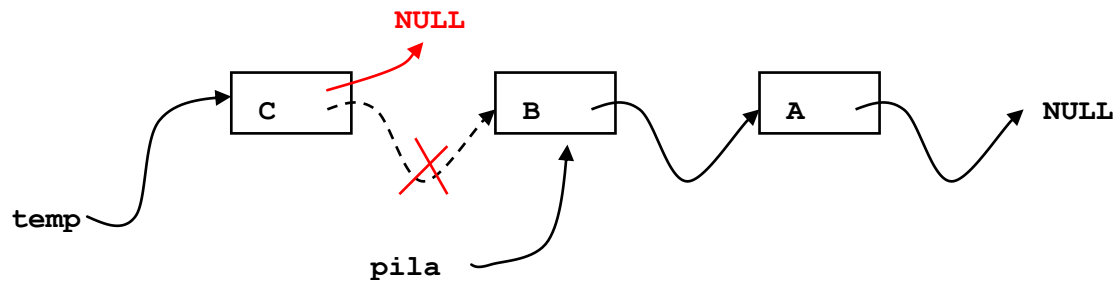


Dopo la chiamata della funzione pop() avremo:

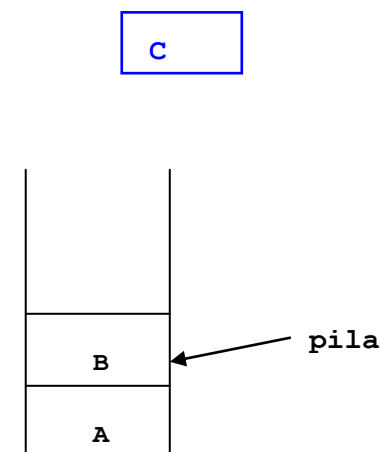
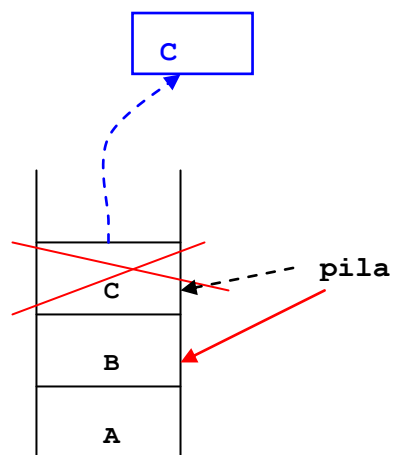
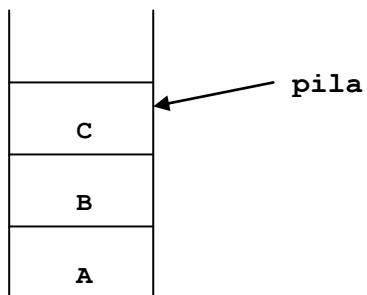
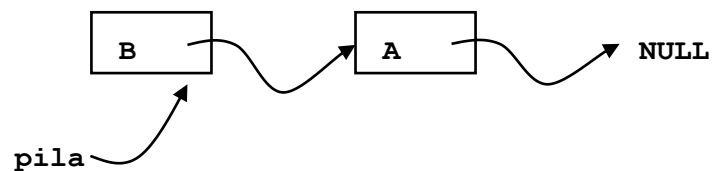
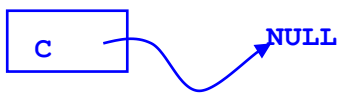
1



2



3



Dove in questo caso la cella "C" è quella restituita dalla funzione!

La funzione pop():

```
struct cella* pop(struct cella **p)
{
    struct cella *temp;
    if (*p == NULL)
        return NULL;

    temp = *p;
    *p = (*p)->next;
    temp->next = NULL;
    return temp;
}
```