

Esame di Programmazione Cl. B (Parte 2)

31 Gennaio 2024 – Turno 1

Ingegneria e Scienze Informatiche A.A. 2023-2024

Si richiede di sviluppare un sistema di gestione delle telecomunicazioni basato sull'utilizzo di liste circolari doppiamente concatenate.

Il sistema prevede la gestione di una lista di messaggi, dove i messaggi vengono inseriti, rimossi e stampati utilizzando le operazioni di inserimento, rimozione e stampa su una lista circolare doppiamente concatenata.

Descrizione della struttura dati:

Utilizzare una struttura dati di tipo lista circolare doppiamente concatenata chiamata **MessageList** per gestire i messaggi.

Ogni messaggio è rappresentato da una stringa di massimo 50 caratteri. La lista deve poter contenere un numero variabile di messaggi.

```
typedef struct Node {
    char message[50];
    struct Node *prev;
    struct Node *next;
} Node;

typedef struct {
    Node *head;
} MessageList;
```

Prototipi delle funzioni (fornite nel file *telecom.h*):

- **void initializeMessages(MessageList *messageList):** Inizializza la lista dei messaggi. – 1 Punto
- **int isMessageEmpty(MessageList *messageList):** Restituisce 1 se la lista dei messaggi è vuota, altrimenti restituisce 0. – 3 Punti
- **void sendMessage(MessageList *messageList, const char *message):** Inserisce un messaggio nella lista. – 3 punti
- **void receiveMessage(MessageList *messageList):** Rimuove e stampa il messaggio in testa alla lista. Stampa un messaggio di errore se la lista è vuota. – 3 punti
- **void printMessages(MessageList *messageList):** Stampa tutti i messaggi presenti nella lista. – 3 punti

- **`void deleteAllMessages(MessageList *messageList)`**: Rimuove tutti i messaggi presenti nella lista. – 3 punti

Il file **telecom.h** è strutturato nel seguente modo:

```
#ifndef TELECOM_H
#define TELECOM_H

typedef struct Node {
    char message[50];
    struct Node *prev;
    struct Node *next;
} Node;

typedef struct {
    Node *head;
} MessageList;

void initializeMessages(MessageList *messageList);
int isEmpty(MessageList *messageList);
void sendMessage(MessageList *messageList, const char *message);
void receiveMessage(MessageList *messageList);
void printMessages(MessageList *messageList);
void deleteAllMessages(MessageList *messageList);

#endif // TELECOM_H
```

Test del programma:

Nel file **telecom_main.c**, utilizzare le funzioni implementate per testare il sistema di gestione delle telecomunicazioni.

Inviare, ricevere, stampare e cancellare messaggi, gestendo correttamente situazioni di lista vuota.

Il file da consegnare è esclusivamente il file **telecom.c**, dove devono essere implementate le funzioni richieste per la gestione delle liste circolari doppiamente concatenate.

In caso di utilizzo della funzione `strcpy_s` se ne ricorda la sintassi:

`strcpy_s(char *dest, rsize_t destsz, const char *src)`;

Dove:

- **`dest`**: Puntatore alla stringa di destinazione (buffer).
- **`destsz`**: Dimensione del buffer di destinazione.
- **`src`**: Puntatore alla stringa di origine da copiare nel buffer di destinazione.

`strcpy_s` restituisce un valore diverso da zero, nel caso di errore durante la copia della stringa.