Esercizi sugli Stack

Roberto Borelli

22 Novembre 2023

Consideriamo la seguenta implementazione tramite array di uno stack:

```c
#include <stdio.h>
#define N 1000// dimensione massima stack
#define NULL_ELEMENT -1
int TOS = 0;
int stack[N];

void inizializzaStack(){
    // inizializzo ogni poszione
    for(int i = 0; i < N; i++){
        stack[i] = NULL_ELEMENT;
    }
    TOS = 0;
}

void push(int el){
    if(TOS == N){
        // stampa messaggio di errore
    }else{
        // lo stack (pila) non e' pieno quindi inseriamo el nello stack
        stack[TOS] = el;
        TOS++;
    }
}

// prende l'lemento in cima alla pila, lo rimuove e lo restituisce
int pop(){
    if(TOS > 0){
        TOS --;
        int el = stack[TOS];
        stack[TOS] = NULL_ELEMENT;
        return el;
    }else{
        return NULL_ELEMENT; // lo stack era gia vuoto
    }

}
```

1. **Implementazione di funzioni base**:
   Scrivere le funzioni `isFull()` per controllare se lo stack è pieno e `isEmpty()` per verificare se è vuoto. Scrivere la funzione `top()` che restituisca l'lemento in cima dello stack ma senza rimuoverlo. Utilizzare queste funzioni nel main per testare lo stack in condizioni diverse.

2. **Stampa dello stack**:
   Scrivere una funzione `printStack()` che stampi tutti gli elementi presenti nello stack al momento della chiamata. Utilizzare questa funzione nel main per visualizzare lo stato dello stack dopo ogni `push()` o `pop()`.

3. **Svuotamento dello stack**:
   Scrivere una funzione `clearStack()` che svuoti completamente lo stack, rimuovendo tutti gli elementi. *Vincolo*: per accedere allo stack utilizzare solo le funzioni `isFull()`, `isEmpty()`, `push()`, `pop()` o `top()` ed è vietato accedere direttamente ad `stack[i]` per qualunque valore di `i`.

4. **Svuotamento dello stack fino ad un certo elemento**:
   Scrivere una funzione `clearStackUntil()` che accetti un valore `val` come argomento, parta dalla cima dello stack e rimuova elementi fino a quando `val` non si trovi in cima. Se `val` non è presente nello stack, lo stack viene svuotato tutto. *Vincolo*: per accedere allo stack utilizzare solo le funzioni `isFull()`, `isEmpty()`, `push()`, `pop()` o `top()` ed è vietato accedere direttamente ad `stack[i]` per qualunque valore di `i`.