

Fondamenti di Informatica T-1 modulo 2

Laboratorio 09: **stack e code**

1

Esercizio 1: Stack

Conversione decimale-binario mediante stack

- Si realizzi un programma che, **utilizzando uno stack**, converta un numero decimale nella corrispondente rappresentazione binaria
- Il risultato deve essere **salvato in una stringa della giusta dimensione**
 - Utilizzare il logaritmo in base 2 (con qualche aggiustamento) per conoscere il numero di cifre necessarie

2

Esercizio 1 - Requisiti

Funzioni da realizzare

- `int calculateSpace(int num)`
 - Calcola lo spazio necessario per memorizzare il risultato, a partire dal numero decimale
- `void calculate(stack* s)`
 - **Effettua la conversione ricorsivamente, mettendo il risultato sullo stack**
- `void consume(stack* s, char* result)`
 - Prende dati dallo stack e produce il risultato sotto forma di stringa
 - A fine calcolo, lo stack deve tornare a essere vuoto
- `char* convertToBinary(int num, stack* binaryStack)`
 - Funzione per il cliente (invoca le altre funzioni e restituisce la stringa risultato)

3

Esercizio 1 – Pseudo-codice

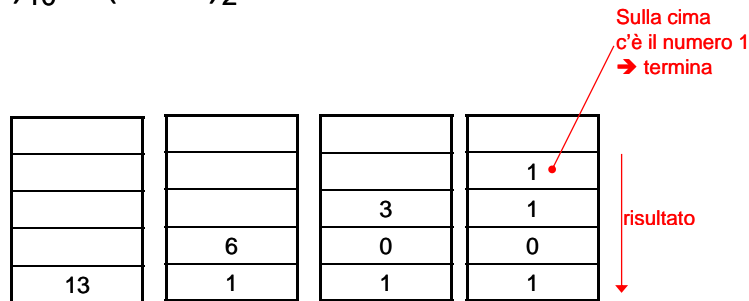
1. Inserisci il numero decimale nello stack
2. Estrai l'elemento `e1` in cima allo stack
3. Se `e1` è un bit (0 o 1) **termina**
4. Altrimenti inserisci nello stack il resto della divisione intera tra `e1` e 2
5. Inserisci nello stack il risultato della divisione intera tra `e1` e 2
6. Ritorna al punto 2

NOTA: il risultato della conversione è proprio la sequenza di dati dalla cima al fondo nello stack

4

Esercizio 1 - Esempio

$$(13)_{10} = (1101)_2$$



5

Esercizio 2: Stack

Riconoscimento palindromi mediante stack

- Si realizzi un programma che, **utilizzando (due) stack**, prenda in input una stringa e valuti se essa è palindroma o meno
- Una stringa è palindroma se la seconda metà è esattamente speculare rispetto alla prima metà
- Esempi:
 - “radar” è palindroma
 - “non” è palindroma
 - “nono” non è palindroma
 - “onorarono” è palindroma

6

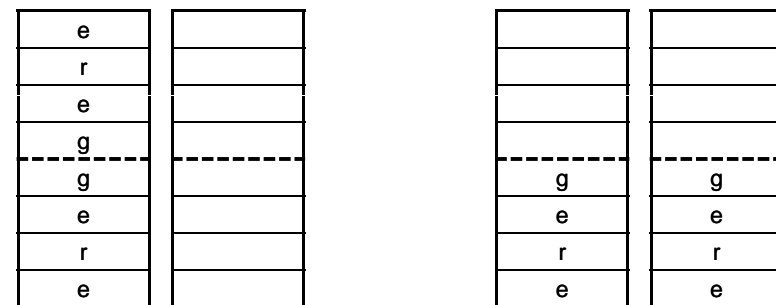
Esercizio 2 – Linee Guida di Soluzione

- **Inserire tutta la stringa in uno stack**
- Prendere un **secondo stack** e “riversare” metà del primo stack nel secondo
- Se i due stack sono a questo punto uguali (stessi elementi nello stesso ordine), la stringa è palindroma
 - E se la stringa è di lunghezza dispari? Serve un piccolo accorgimento...

7

Esercizio 2 - Esempio

Esempio: “ereggere”



8

Esercizio 3: Code FIFO

Coda FIFO per la playlist di un jukebox

- Si realizzi un programma che, **utilizzando una coda FIFO**, simuli la gestione di una playlist
- La coda FIFO deve essere definita per il tipo di dato **Song**, composto da **nome dell'autore** (al più 20 caratteri), **nome della canzone** (al più 20 caratteri) e **durata** della canzone in secondi (int)

9

Esercizio 3 - Requisiti

Funzioni da realizzare

- **void** playSong(startQueue* start, endQueue* end)
 - Simula la riproduzione della prossima canzone, stampandone il contenuto sullo standard output
- **void** printPlaylist(startQueue* start, endQueue* end)
 - Stampa tutte le canzoni sullo standard output
 - Alla fine della stampa la coda deve tornare allo stato iniziale (si utilizzi una nuova coda accessoria)
- **int** insertSong(Song el, startQueue* start, endQueue* end)
 - Inserisce una nuova canzone, ma solo se non è già presente nella playlist
 - Alla fine del controllo **la coda deve tornare allo stato iniziale (NON si utilizzi una nuova coda accessoria)**

10

Esercizio 3: Nota

- La funzione **deQueue** è “*distruttiva*”, restituisce gli elementi in coda consumando la coda stessa
- Per iterare lungo gli elementi di una coda mantenendone inalterato il contenuto è possibile
 - **costruire una nuova coda con gli stessi elementi della coda originaria**
 - inserire gli **elementi prelevati in fondo alla coda esaminata** (come fare a capire quando siamo ritornati al primo elemento?)

11