Corso di Laurea in Ingegneria Informatica

Calcolatori — a.a. 2024–2025

Compito del 9 giugno 2025

Cognome e Nome dello studente: _____

Codifica incrementale

*Parole consecutive in un elenco alfabetico (come un indice, o un dizionario) hanno spesso in comune alcune lettere della prima parte della parola (*prefisso*). Questo fatto è utilizzato nella tecnica detta "codifica incrementale" per memorizzare l'elenco in forma compressa (senza perdite). La tabella qui sotto riporta sette parole in ordine alfabetico, insieme alla loro lunghezza (in byte) L. La terza colonna contiene i prefissi condivisi da ciascuna parola con quella che la precede. La codifica incrementale dell'elenco è data nelle ultime tre colonne della tabella. Si noti come di ogni parola sia necessario memorizzare soltanto la parte che segue il prefisso, detta* suffisso*. Tralasciando i due byte per parola necessari per memorizzare le lunghezze di prefisso e suffisso, con i dati dell'esempio il risparmio di memoria è di* $20/55 \times 100 \approx 36\%$*. In media il risparmio ottenibile con questa tecnica di codifica è intorno al* 40%*.*

parola	L(parola)	prefisso	L(prefisso)	L(suffisso)	suffisso
calco	5	— (no parola precedente)	0	5	calco
calcolatore	11	calco	5	6	latore
caldaia	7	cal	3	4	daia
macchina	8	— (no lettere in comune)	0	8	macchina
macedone	8	mac	3	5	edone
macedonia	9	macedon	7	2	ia
maestro	7	ma	2	5	estro
totali	55		20	35	

Assembler ♢ Con riferimento all'esempio in descrizione, siano dati una stringa `wordlist` che riporti, concatenate, n parole in ordine alfabetico crescente e il vettore di n interi `lengths` contenente le lunghezze (in byte) delle parole della stringa. Scrivere un programma in assembly 8086 per codificare `wordlist` nella tripletta (`p_len`,`s_len`,`suffixes`), dove `p_len` e `s_len` sono vettori di n interi contenenti rispettivamente le lunghezze di prefissi e suffissi, e `suffixes` è la stringa degli n suffissi. Il programma deve impiegare almeno una procedura con passaggio di parametri attraverso lo stack. Simulare il funzionamento del programma con i dati dell'esempio.

Macchine logiche ♢ Progettare una macchina sequenziale sincrona che riceva in ingresso ad ogni istante $t = 0, 1, \ldots$ un intero $v(t)$ e produca in uscita la differenza $\delta(t) \doteq v(t) - v(t-1)$ tra ciascun ingresso e quello precedente ($v(-1) \doteq 0$). Dimostrare come la macchina possa servire a ridurre, senza perdita d'informazione, la quantità di bit della sequenza in ingresso, nell'ipotesi che quest'ultima rappresenti segnali (come la voce, o la musica) che variano lentamente nel tempo. Simulare il funzionamento della macchina per la sequenza $v(t) = 172, 174, 176, 179, 177, 180, 181, 178, \ldots$