

(1) 考え方

`comp` 関数は、入力された文字列を走査し、同じ文字が連続する回数をカウントして「回数+文字」という形式で圧縮する関数である。動的にメモリを確保して結果を格納し、異なる文字が現れたときや文字列の末尾に達したときに、現在の連続文字の情報を出力用バッファに書き込む仕組みである。圧縮後の文字列は標準出力に表示され、その長さが戻り値として返される。また、動的メモリを使用しているため、最後に確保したメモリを解放してリソース管理を適切に行う設計となっている。`decomp`関数は、入力された圧縮形式の文字列を解釈し、「回数+文字」の情報を元に元の文字列を復元する関数である。入力を走査しながら連続する数字部分を解析して回数を取得し、その後の文字を指定された回数だけ出力用バッファに書き込む仕組みである。形式が不正である場合はエラーメッセージを出力し、処理を中断する。展開後の文字列は標準出力に表示され、動的に確保したメモリは最後に解放される。入力の形式チェックと動的メモリ管理が適切に行われる設計となっている。

(2) 注意点

`malloc` で動的に確保したメモリを忘れずに `free` で解放している。メモリ領域に気をつける必要がある。

(3) 工夫点

圧縮率を戻り値で返している。