Readme file for software-managed compression program:

This program implements the idea suggested in the paper “Improving Code density using compression techniques”. This program emulates a post-compilation analyzer, which examines a program and replaces common sequence of instructions with a single instruction codeword.

**INPUT:** The input to this program is an assembly language program.
**OUTPUT:** This program calculates the compression ratio, using the formula mentioned below:

\[
\text{Compression ratio} = \frac{\text{compressed size}}{\text{original size}}.
\]

Also it writes the compressed code in a file named as `compressed_file` and the dictionary in a file named as `dictionary_file`.

Example of the compression:
How to run the program:

Create an assembly language program
>cc –S prog.c
This will generate an assembly language file prog.s
Compile the program
>cc –o progo prog.c
Run the object code generated
>progo prog.s

The output of program is as follow:

************************** Code Compression Statistics **************************
The input file (prog.s) size: 758
The compressed file (compressed_file) size: 569
The compression percentage: 24.934037
The dictionary for codeword is in file dictionary file
************************** Code Compression Statistics **************************

Result: For Intel Pentium IV architecture, this program does a compression of 25-30% approximately.

P.S: This program takes care that no control instructions is included in the codeword.