

## CS/SE 2340 Homework 5: Compression Program

### Worth 200 points (50 point bonus)

Objective: Gain more experience with MIPS programming, and in using macros.

Overview: In this program you will implement a very simple compression algorithm, RLE, as described in Wikipedia:

**Run-length encoding (RLE)** is a very simple form of lossless data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs.

For example, 'AABBBC' would be encoded as: 'A2B3C1'

### Instructions

---

Macro File: Create macros to print an int, print a char, print a string, get a string from the user, open file, close file, read file, and allocate heap memory. You can use more macros than these if you like.

Main Program File:

1. Allocate 1024 bytes of dynamic memory and save the pointer to the area.
2. The main program is a loop in which you ask the user for a filename. If they enter nothing for the filename, exit the program. Otherwise:
  - a. Open the file for reading. If the file does not exist, print an error message and terminate the program.
  - b. Read the file into an input buffer space of 1024 bytes.
  - c. Close the file.
  - d. Invoke the print string macro to output the original data to the console.
  - e. Call the compression function. Save the size of the compressed data in memory.
  - f. Call a function to print the compressed data.
  - g. Call the uncompress function, which uncompresses to the console.
  - h. Print the number of bytes in the original and compressed data.
3. The compression function implements the RLE algorithm above and stores the compressed data in the heap. Before the function call, set \$a0 to the address of the input buffer, set \$a1 to the address of the compression buffer, set \$a2 to the size of the original file. The function should "return" the size of the compressed data in \$v0.
4. The uncompress function will repeat characters as indicated in the compressed file. For example, if the compressed file is 'A2B3C1' it will print

'AABBBC'. Note that this is different from the print string macro which just uses a syscall to print a string.

What to turn in:

- Upload your .asm files to eLearning, zipped together. Keep your macros in a separate .asm file from the main program .asm file.

Grading Rubric:

Points	Element
50	Macros as described above
40	Function to compress input file
40	Function to uncompress data to the console
30	Function to output compressed data
30	Main program with loop
10	Comments, good use of whitespace
200	Total

Below is sample output for file error:

```
Please enter the filename to compress or <enter> to exit: tmp.txt

Error opening file. Program terminating.
-- program is finished running --
```

Below is sample output for file hello.txt, showing what happens when the user hits enter at the filename prompt:

```
Please enter the filename to compress or <enter> to exit: hello.txt

Original data:
hello
Compressed data:
h1e1l2o1
Uncompressed data:
hello
Original file size: 5
Compressed file size: 8
Please enter the filename to compress or <enter> to exit:

-- program is finished running --
```

```
Please enter the filename to compress or <enter> to exit: hello_art.txt

Original data:
~( <->) (<->)

Compressed data:
2_3 1_3 9_1_2 2_1_2 12_2 6_2 12_1_2 7_1_3_1_1_1
1 1/1 3|1 3\1 3_4 1|1 2|1 1|1 2|1 3_4 3/1 2\1 4/1 2\1_11|1 2|1 4_2|1 1_1/1|1 1|1
1/1 4~1 4\1_1/1 1_2 1\1|1 2|1 1|1 2|1 2/1 2_1 1\1 2\1 3\1/1\1/1 3/1 2_1 1\1_1 2_2 1\1 2|1 3/1 1_2 1|1 1
1\1 4Y1 4/1\1 2_3/1|1 2|1_1|1 2|1_1(1 2<1_1>1) 1|1 2\1 8(1 2<1_1>1) 1|1 2|1 1\1/1 2|1_2/1 1/1_1/1 1|1 2\1
1 1\1_3|1_1 2/1 2\1_3 2>1_4/1_4/1\1_4/1 4\1_2/1\1 2/1 1\1_4/1|1_2|1 2|1_4/1_1_4 1|1 2_2
1 7\1/1 7\1/1 26\1/1 25\1/1 2\1/1

Uncompressed data:
~( <->) (<->)

Original file size: 467
Compressed file size: 462
Please enter the filename to compress or <enter> to exit: hello.txt

Original data:
hello
Compressed data:
h!e1l2o1
Uncompressed data:
hello
Original file size: 5
Compressed file size: 8
Please enter the filename to compress or <enter> to exit:

-- program is finished running --
```

hello.txt  
hello\_art.txt

<http://patorik.com/software/taag/#p=display&f=Graffiti&t=Hello%20World!>

Implement a more complex lossless compression algorithm of your choice. Document any sources you use in your code.