1. Include necessary headers such as #include<stdio.h>
2. Define constants for maximum word and line lengths; such as, #define MAX\_WORD\_LENGTH
3. Function to check if a word can fit into the line Strlen can be used for it

# Function to print the justified line:

void printJustifiedLine(char \*\*words, int word\_count, int max\_line\_length, bool is\_last\_line) {

// Calculate total characters and spaces needed

// Print words with spaces distributed evenly

}

# Function to split and process a hyphenated word across lines:

void processHyphenatedWord(char \*word, char \*\*words, int \*word\_count, int

\*current\_line\_length, int max\_line\_length) {

// Split word at hyphen and process each part

}

# Function to process the file and justify text:

void processFile(const char \*filename, int max\_line\_length) {

// Open file and read words

// Handle word fitting and hyphenation

// Print justified lines

// Free allocated memory

}

1. **Main function to handle command-line arguments and call** processFile int main(int argc, char \*argv[]) {

if (argc …) {

….

}

int max\_line\_length = atoi(argv); if (…) {

…..

}

return 1;

processFile(argv, max\_line\_length); return 0;

}