```c
#include <stdint.h>
#include <stdio.h>

// Written by Taylor Cowley and Andrew Okazaki

FILE *fileptr;          // Pointer to input file
FILE *outFILE;          // Pointer to output file
char *buffer;           // the buffer for reading chars in
long filelen;           // the lenth of the file

// Main.
int main(int argc, char **argv){

    // A simple "testing to make sure you gave the right filenames"
    int32_t i;
    for(i = 0;i<argc; i++){
        printf("%s\n", argv[i]);
    }


    // Prints out the first 10 bytes of the file to make sure it's good
    unsigned char buffer[10];           // reading 10 bytes at a time?
    fileptr = fopen(argv[1], "rb");     // Gotta read binary
    fread(buffer, sizeof(buffer),1, fileptr);   // open the file for reading
    int32_t j;
    for(j=0;j<10;j++){                  // read 10 bytes of the file, print hex
        printf("%x ", buffer[j]);
    }

    // Now we go through the file until we reach 'data'
    unsigned int single_byte = 0;
    while(1){
        fread(&single_byte, sizeof(char), 1, fileptr);
        if(single_byte == 'd'){
            fread(&single_byte, sizeof(char), 1, fileptr);
            if(single_byte == 'a'){
                fread(&single_byte, sizeof(char), 1, fileptr);
                if(single_byte == 't'){
                    fread(&single_byte, sizeof(char), 1, fileptr);
                    if(single_byte == 'a'){
                        break;
                    }
                }
            }
        }
    }

    // If we get here, we have reached the data portion of the wav file.
    // Now to get ready to write
    outFILE = fopen(argv[2], "w");  // open the file for writing
    if(outFILE==NULL){
        printf("bad file\n\r");     // error message
        return;
    }

    // We are good
    fprintf(outFILE, "START of our awesome file:\n\n");
```

```c
    // fread will return 0 into yeah when we reach the end of the file
    size_t yeah;
    while((yeah = fread(&single_byte, sizeof(char), 1, fileptr)) != 0){
        fprintf(outFILE, "%u, ", single_byte);  // Just write the byte to the new file
    }

    // Close both of the files
    fclose(outFILE);
    fclose(fileptr);
    return 0;
}
```