```
// example.c

//========================
// How to compile on Athena
//========================
// gcc -o example example.c

//============
// Getting Help
//============
// if you wonder about any function or library used here,
// you can usually just type man FUN or man LIB at prompt

#include <stdio.h>
#include <limits.h>
//======
// Macros
//======
// LONG_MAX is defined in <limits.h>
#define INFINITY LONG_MAX
#define max(A, B) ((A) > (B) ? (A) : (B))

#define DEBUG 1

//----------------
// Arbitrary Size Limits
//----------------
// We don't want to do any memory management (using malloc & friends)...
// So we have to arbitrarily define the size of our arrays at compile-time
#define WORDLEN 50 /* max number of characters in a word */
#define SEQLEN 200 /* max number of words in a text */
#define FN_LEN 256 /* max number of characters in a filename */
int main(int argc, char *argv[]) {
  //======
  // Arrays (the easy static way)
  //======
  // - a string is an array of chars, its end being signaled by a '\0'
  char word[WORDLEN];
  // 1****************************************1
  word[0] = 'h'; word[1] = 'i'; word[2] = '\0';
  if (DEBUG) printf("1.␣word␣is␣'%s'\n", word);
  // 1****************************************1
  // - an array of strings is actually an array of arrays of chars
  char seq[WORDLEN][SEQLEN];
  // - an array of ints
  int lenseq[SEQLEN];
```

```
//=====================
// Command-Line Arguments
//=====================
// argc gives you the number of arguments (including the executable name)
// argv is an array of the actual arguments
char *filename;
if (argc!=2) {
  printf("USAGE:␣./example␣filename.txt\n");
  exit(1); // something other than 0 indicate an error
} else {
  filename = argv[1];
}
// 2**************************************2
if (DEBUG) printf("2.␣filename␣is␣'%s'\n", filename);
// 2**************************************2

//==================
// Reading from a File
//==================
FILE *in_file;
in_file = fopen(filename, "r");
if (in_file==NULL) {
  printf("Sorry,␣filename␣'%s'␣could␣not␣be␣open.\n", filename);
  exit(1);
}
//----------------------------------------
// Read the content as a sequence of words
//----------------------------------------
int i = 0;
while (!feof(in_file) && i<SEQLEN) {
  if (fscanf(in_file, "%s", word) == 1) {
    strcpy(seq[i++], word);
  }
}
fclose(in_file);
int total = i;

for (i=0; i<total; i++) {
  lenseq[i] = strlen(seq[i]);
}

//================
// Writing to a file
//================
FILE *out_file;
```

```
  char out_filename[FN_LEN];
  strcpy(out_filename, "out-");
  strcat(out_filename,filename);
  out_file = fopen(out_filename, "w");
  if (out_file==NULL) {
    printf("Could␣not␣write␣to␣filename␣'%s'", out_filename);
    exit(1);
  }
  for (i=0; i<total; i++) {
    fprintf(out_file, "%2d.␣%s␣(%d)\n", i, seq[i], lenseq[i]);
  }
  fclose(out_file);
}
```