

```

1  #include "decl.h"
2  #include <stdio.h>
3  #include <string.h>
4  #include <stdlib.h>
5
6  /**
7   * Write the player's information from the struct to the file and
8   * read pre-existing data into a buffer to be displayed
9   * @param info The struct containing the player's info
10  */
11 void readPlayersInformation(struct PlayerInfo *info)
12 {
13     /*Open the file for writing*/
14     FILE *outFile = fopen("summary.txt", "a");
15     fprintf(outFile, "%s %s %d\n", info->name, info->date, info->numGenerations);
16
17     /*
18      I close the file here only to open it again which may seem redundant,
19      but the reason is a file cannot be opened for reading if it hasn't
20      been created. However, opening a file for appending if it doesn't exist will
21      create
22      it. Thus, I ensure that the file will be written to regardless.
23     */
24     fclose(outFile);
25
26     char *entries[BUFFER_SIZE];
27     char buffer[BUFFER_SIZE];
28     int line = 0;
29     outFile = fopen("summary.txt", "r");
30
31     /*Read the file line by line until EOF is reached*/
32     while (fgets(buffer, BUFFER_SIZE, outFile))
33     {
34         /*
35          Allocate memory on the heap for each "string" by malloc-ing
36          enough bytes to fit it as well as the terminating character.
37          Note: sizeof(char) is redundant since a char occupies 1 byte
38          but it increases readability and makes it explicitly clear to
39          anyone reading the code.
40         */
41         entries[line] = malloc(sizeof(char) * (strlen(buffer) + 1));
42
43         /*After allocating enough memory, store the string in the current index*/
44         strcpy(entries[line++], buffer);
45     }
46     fclose(outFile);
47     displayGameSummary(entries, line);
48 }

```