Princeton University
COS 217: Introduction to Programming Systems
C Text File Handling

Opening a Text File for Writing

```
#include <stdio.h>
FILE *psFile;
psFile = fopen("filename", "w");
```

Open *filename* for writing.
Return the address of a FILE structure (or NULL).

Note: stdout and stderr are predefined variables of type FILE*

Writing Data to a Text File

Character:

```
iStatus = fputc(iChar, psFile);
iStatus = putc(iChar, psFile);  /* May be a macro */
iStatus = putchar(iChar);       /* May be a macro */
```

Write iChar to psFile (or stdout). Return iChar (or EOF).

String:

```
iStatus = fputs(pcString, psFile); /* Omits '\0' */
iStatus = puts(pcString);          /* Replaces '\0' with '\n' */
```

Write pcString to psFile (or stdout). Return a non-negative number (or EOF).

Formatted data:

```
iStatus = fprintf(psFile, "%d", i);
iStatus = printf("%d", i);
```

Convert i to a sequence of ASCII digits. Write those digits to psFile (or stdout).
Return the number of digits written (or EOF).

See King pp. 487-492 for fprintf() conversion specification for each data type.

Opening a Text File for Reading

```
#include <stdio.h>
FILE *psFile;
psFile = fopen("filename", "r");
```

Open *filename* for reading. Return a pointer to a FILE structure (or NULL).

Note: stdin is a predefined variable of type FILE*.

Reading Data from a Text File

Character:

```
iChar = fgetc(psFile);
iChar = getc(psFile);  /* May be a macro */
iChar = getchar();     /* May be a macro */
```

Read an ASCII code from psFile (or stdin). Return the ASCII code (or EOF).

Line:

```
pcStatus = fgets(pcString, iBufferSize, psFile);
   /* Appends '\0' */
pcStatus = gets(pcString);
   /* Replaces '\n' with '\0' */
   /* Dangerous:  May corrupt memory */
```

Read a line from psFile (or stdin) into the memory at address pcString. Return pcString (or NULL).

Formatted data:

```
iStatus = fscanf(psFile, "%d", &i);
iStatus = scanf("%d", &i);
```

Read a sequence of ASCII digits from psFile (or stdin); stop at the first non-digit character. Convert the sequence of digits to an integer. Assign the integer to memory at address &i. Return the number of values read (or EOF).

See King pp. 492-496 for fscanf() conversion specifications for each data type.

Closing a Text File

```
iStatus = fclose(psFile);
```

Close psFile, and return 0 (or EOF).