EXP 25: Construct a C program to implement the I/O system calls of UNIX (fcntl, seek, stat, opendir, readdir)

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/stat.h>
#include <dirent.h>
#include <string.h>
#include <errno.h>

int main() {
   const char *filename = "sample.txt";

   // Create or open the file
   int fd = open(filename, O_RDWR | O_CREAT, 0644);
   if (fd < 0) {
      perror("open");
      return 1;
   }

   // Use fcntl to get file status flags
   int flags = fcntl(fd, F_GETFL);
   if (flags < 0) {
      perror("fcntl - F_GETFL");
   } else {
      printf("File flags for %s: %d\n", filename, flags);
   }
```

```
// Use lseek to move the file offset
off_t offset = lseek(fd, 0, SEEK_END);
if (offset < 0) {
    perror("lseek");
} else {
    printf("Current end of file offset: %ld\n", (long)offset);
}

// Use stat to get file information
struct stat fileStat;
if (stat(filename, &fileStat) < 0) {
    perror("stat");
} else {
    printf("\nFile: %s\n", filename);
    printf("Size: %ld bytes\n", fileStat.st_size);
    printf("Permissions: %o\n", fileStat.st_mode & 0777);
    printf("Last modified: %ld\n", fileStat.st_mtime);
}

close(fd); // Close file

// Use opendir and readdir to list files in current directory
printf("\nListing files in current directory:\n");
DIR *dir = opendir(".");
if (!dir) {
    perror("opendir");
    return 1;
}
```

```
    struct dirent *entry;
    while ((entry = readdir(dir)) != NULL) {
        printf(" - %s\n", entry->d_name);
    }

    closedir(dir);

    return 0;
}
```

Sample Output

```
File flags for sample.txt: 2
Current end of file offset: 0

File: sample.txt
Size: 0 bytes
Permissions: 644
Last modified: 1715100000

Listing files in current directory:
 - .
 - ..
 - sample.txt
 - main.c
 - a.out
```

```
open: Permission denied


=== Code Exited With Errors ===
```