TEST data for EXTENT file system
We are testing extent file system by creating two test files and then getting their metadata information.
First file has 8 byte strings on 64 lines and 2nd file has 7 byte strings on 64 line again.
We are testing if the File system we created is EXTENT, by displaying the “Type” defined in stat.h

```
$ fsTest
**************** Starting test for file1 *******************
Writing abcdef to line0 of file1
.
.
.
Writing abcdef to line64 of file1
File size i= 520 <- Larger than 512, so it should require two blocks
File type = 4 <- T_EXTENT file type is defined as 4
File system device = 1
inode num = 20
Number of links = 1
Block Addresses:
  Pointer 0's address = 652 with size = 2 <-The EXTENT file properly expanded onto a second block
  Pointer 1's address = 0 with size = 0 <-The EXTENT file didn’t use a second direct pointer for the second block
  Pointer 2's address = 0 with size = 0
  Pointer 3's address = 0 with size = 0
  Pointer 4's address = 0 with size = 0
  Pointer 5's address = 0 with size = 0
  Pointer 6's address = 0 with size = 0
  Pointer 7's address = 0 with size = 0
  Pointer 8's address = 0 with size = 0
  Pointer 9's address = 0 with size = 0
  Pointer 10's address = 0 with size = 0
  Pointer 11's address = 0 with size = 0
  Pointer 12's address = 0 with size = 0
Number of blocks = 0
************* Starting test for file2 ***************
Writing hello to line0  of file2
.
.
.
Writing hello to line64  of file2
File size i= 455 <- Smaller than 512, so it should require one block
File type = 4 <- T_EXTENT file type is defined as 4
File system device = 1
inode num = 21
Number of links = 1
Block Addresses:
 File2: Pointer 0's address = 654 with size = 1 <-The EXTENT file properly allocated only one block
 File2: Pointer 1's address = 0 with size = 0
 File2: Pointer 2's address = 0 with size = 0
 File2: Pointer 3's address = 0 with size = 0
 File2: Pointer 4's address = 0 with size = 0
 File2: Pointer 5's address = 0 with size = 0
 File2: Pointer 6's address = 0 with size = 0
 File2: Pointer 7's address = 0 with size = 0
 File2: Pointer 8's address = 0 with size = 0
 File2: Pointer 9's address = 0 with size = 0
 File2: Pointer 10's address = 0 with size = 0
 File2: Pointer 11's address = 0 with size = 0
 File2: Pointer 12's address = 0 with size = 0
Number of blocks = 0
***********Testing Extent File system completed!*************
$ ls
.            1 1 512
..           1 1 512
README       2 2 2290
cat          2 3 14458
echo         2 4 13331
forktest     2 5 8177
grep         2 6 16014
init         2 7 14208
kill         2 8 13375
ln           2 9 13297
ls           2 10 16317
mkdir        2 11 13396
rm           2 12 13373
sh           2 13 24825
stressfs     2 14 14291
usertests    2 15 67223
wc           2 16 15144
fsTest       2 17 17820
zombie       2 18 13041
console      3 19 0
mytestFile.txt 4 20 520 <- ls is properly seeing our generated EXTENT files
mytestFile2.tx 4 21 455
$ wc mytestFile.txt
65 66 520 mytestFile.txt <- wc is properly accessing our generated EXTENT files
$ wc mytestFile2.tx
65 66 455 mytestFile2.tx
$
```

TEST data for lseek() system call
We test lseek() by writing 64 characters to a file, calling lseek() to change the offset to 26, then writing 36 characters to the file. The 36 characters overwrite the characters in the original file, demonstrating that the offset works.
We also verify that lseek() fails if directed to change the offset to a value outside of the current file using either a negative number or a number that is larger than the size of the file.