Part1

Firstly I initialized required variables.I store ./appendMeMore argument in controlArgument for controlling execute of program.If user enters any other thing, the program will not executed.Then with controlFlag2, I control if user enters ‘x’ argument.If flag is 1, then program executes for x argument part.It opens file without APPEND flag and uses lseek(fd, 0, SEEK_END) before every byte write .If flag is 1, then program enters APPEND flag open.I write every byte in loop.Created bytes as random characters.

```
-rwxrwxr-x 1 karacete karacete   17120 Mar 30 22:31 appendMeMore
-rw-rw-r-- 1 karacete karacete    1620 Mar 30 22:49 appendMeMore.c
-rw-rw-r-- 1 karacete karacete 2000000 Mar 30 22:34 f1
-rw-rw-r-- 1 karacete karacete 1033348 Mar 30 22:50 f2
```

I created f1 file without x argument and it’s size on executing ./appendMeMore f1 1000000 & ./appendMeMore f1 1000000 this command becomes 2 mb.

I created f2 file with x argument and it’s size on executing ./appendMeMore f2 1000000 x & ./appendMeMore f2 1000000 x this commane becomes less than 2mb.

I think may be writing with lseek I may be writing on existing characters.So the file is smaller than file from opened APPEND flag.

Part2

Firstly I created dup() function. fcntl(oldFd,F_DUPFD,0) equal to dup() function.This function creates a new file descriptor that refers to the same open file or socket as the original file descriptor. If we make fd2 = fcntl(oldFd,F_DUPFD,0) fd2 shares open file of oldFD.

Then I created dup2() function. close(newFd) fcntl(oldFd, F_DUPFD, newFd) this two function does same operation as dup2() function.Firstly closes newFd then duplicates oldFd descriptor, and returns a new file descriptor that refers to the same open file description as the original to the newFd.But in dup2() if two fd’s are same and fcntl(oldFd,F_GETFL) returns -1 than this function must return EBADF. In an if statement I controlled that situation and returned error.

```
karacete@karacete-GL553VD:~/Desktop/SystemPrograming$ ./part2
***********Before dup***********
First file content: elma
Second file content: ayva

***********After dup***********
First file content: ayva
Second file content: ayva


***********Before dup2***********
First file content: elma
Second file content: ayva

***********After dup2***********
First file content: ayva
Second file content: ayva
karacete@karacete-GL553VD:~/Desktop/SystemPrograming$
```

Firstly I writed one file elma and other file ayva. Then I tried both dup() and dup2() for fd = dup(fd2) and dup2(fd2,fd) and here is outputs.

Part3

Firstly I created two sentences.This is sentence and This is other sentence.I write them in two text files.Then I maked dup2 and make lseek.Then with firstFileOffset = lseek(fd,0,SEEK_CUR), I take both files offset values.With that I see both file offset showed same number.This is proof of duplicated file descriptors share a file offset value and open file.