CSC 415.03 File System Project (File System Milestone One)

Group Name:
Geek Squad

The GitHub Repository Name That We'd Like You To Grade:
EssaHusary

GitHub Repository Link:
CSC415-2023-Spring/csc415-filesystem-EssaHusary: csc415-filesystem-EssaHusary created by GitHub Classroom

Group Members:

Name	Student ID
Mahdi Hassanpour	922278744
Essa Husary	917014896
Jinwei Tang	922350439
Amit Bal	922832232

1. A dump (use the provided HexDump utility) of the volume file that shows the VCB, FreeSpace, and complete root directory.
2. A description of the VCB structure
3. A description of the Free Space structure
4. A description of the Directory system
5. A table of who worked on which components
6. How did your team work together, how often you met, how did you meet, how did you divide up the tasks.
7. A discussion of what issues you faced and how your team resolved them.

<u>The dump of the volume file that shows the VCB, Free Space, and complete root directory:</u>

<u>VCB DUMP</u>:

```
student@student-VirtualBox:~/Documents/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.lin
ux SampleVolume --count 1 --start 1
Dumping file SampleVolume, starting at block 1 for 1 block:

000200: 00 02 00 00 4B 4C 00 00  38 4C 00 00 00 00 00 00 | ....KL..8L......
000210: 16 39 79 34 91 23 00 00  06 00 00 00 01 00 00 00 | .9y4◆#..........
000220: 05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000230: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000240: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000250: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

000300: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000310: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000320: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000330: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000340: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000350: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000360: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000370: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000380: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000390: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

student@student-VirtualBox:~/Documents/csc415-filesystem-EssaHusary$
```

<u>Analysis of the dump:</u>

Addresses 000 200 - 000 203 contain the value 00 00 02 00 (reversed from little endian). In decimal, this value is 512 which is the field in our VCB called “sizeOfBlocks” which is of type int and stores the size of the blocks of the volume.

Addresses 000 204 - 000 207 contain the value 00 00 4C 4B (reversed from little endian). In decimal, this value is 19,531 which is the field in our VCB called “numOfTotalBlocks” of type int and stores the number of blocks in the volume.

Addresses 000 208 - 000 20B contain the value 00 00 4C 38 (reversed from little endian). In decimal, this value is 19,512 which is the field in our VCB called “numOfFreeBlocks” of type int and stores the number of free blocks in the volume.

Addresses 000 210 - 000 217 contain the value 00 00 23 91 34 79 39 16 (reversed from little endian). In decimal, this value is 39,106,557,589,782 which is the field in our VCB called “magicNumber” of type u_int64_t (since it’s so large) and stores the magic number for the volume.

Addresses 000 218 - 000 21B contain the value 00 00 00 06 (reversed from little endian). In decimal, this value is 6 which is the field in our VCB called “startingBlockOfRootDirectory” of type int and stores the starting block of the root directory.

Addresses 000 21C - 000 21F contain the value 00 00 00 01 (reversed from little endian). In decimal, this value is 1 which is the field in our VCB called “startingBlockOfFreeSpace” of type int and stores the starting block of the free space map.

Addresses 000 220 - 000 223 contain the value 00 00 00 05 (reversed from little endian). In decimal, this value is 5 which is the field in our VCB called “lengthOfFreeSpaceMap” of type int and stores the length of the free space map.

The rest of the values are 00 as those bytes are just the result of mallocing space that’s more than enough (but appropriate) to contain the VCB.

FREESPACE DUMP:

THE HEXDUMP IS ON THE NEXT PAGE

```
student@student-VirtualBox:~/Documents/csc415-filesystem-EssaHusary$ ./Hexdump/h
exdump.linux SampleVolume --count 5 --start 2
Dumping file SampleVolume, starting at block 2 for 5 blocks:

000400: 00 00 1F FF FF FF FF FF  FF FF FF FF FF FF FF FF | ...◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000410: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000420: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000430: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000440: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000450: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000460: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000470: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000480: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000490: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0004F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

000500: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000510: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000520: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000530: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000540: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000550: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000560: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000570: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000580: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000590: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0005F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

```
000600: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000610: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000620: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000630: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000640: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000650: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000660: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000670: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000680: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000690: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0006F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

000700: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000710: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000720: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000730: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000740: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000750: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000760: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000770: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000780: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000790: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0007F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

```
000800: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000810: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000820: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000830: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000840: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000850: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000860: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000870: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000880: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000890: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0008F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

000900: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000910: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000920: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000930: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000940: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000950: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000960: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000970: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000980: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000990: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
0009F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

```
000A00: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A10: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A20: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A30: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A40: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A50: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A60: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A70: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A80: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000A90: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AA0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AB0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AC0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AD0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AE0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000AF0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

000B00: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B10: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B20: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B30: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B40: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B50: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B60: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B70: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B80: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000B90: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BA0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BB0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BC0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BD0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BE0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000BF0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

```
000C00: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C10: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C20: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C30: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C40: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C50: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C60: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C70: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C80: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000C90: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CA0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CB0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CC0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CD0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CE0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000CF0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

000D00: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D10: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D20: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D30: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D40: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D50: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D60: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D70: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
000D80: FF FF FF FF FF FF FF FF  FF E0 00 00 00 00 00 00 | ◆◆◆◆◆◆◆◆◆◆.......
000D90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DD0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000DF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

student@student-VirtualBox:~/Documents/csc415-filesystem-EssaHusary$
```

<u>Analysis of the dump:</u>

Addresses 000 400 - 000 402 contain the hex values 00 00 1F, which in binary is 0b00000000 00000000 00011111. This represents the fact that the VCB, which occupies 1 block, the free space, which occupies 5 blocks, and the root directory, which occupies 13 blocks, are all marked used, with a total of 19 used blocks. We've defined 0 as used and 1 as free. All of the other addresses, from 000 402 - 000 D88 are all free, containing values of FF in hex and 11111111 in binary since we haven't used them yet. Address 000 D89 contains a value of E0 in hex and 11100000 in binary where the 0s simply mean that because we only care about 19,531 blocks, we don't care to assign the blocks/bits after the final 1 any values. Addresses 000 D8A - 000 DFF contain values of 00 or 00000000 in binary as these bytes are just extra space as a result of mallocing space that's more than enough (but appropriate) to contain the free space map.

ROOT DIRECTORY DUMP:

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 7
Dumping file SampleVolume, starting at block 7 for 1 block:

000E00: 2E 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E20: 0A 00 00 00 00 00 00 00  56 1C 32 64 00 00 00 00 | ........V.2d....
000E30: 56 1C 32 64 00 00 00 00  00 00 00 00 FF FF FF FF | V.2d........????
000E40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000E50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000E60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000E70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
000E80: 2E 2E 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EA0: 80 00 00 00 00 00 00 00  56 1C 32 64 00 00 00 00 | ?.......V.2d....
000EB0: 56 1C 32 64 00 00 00 00  00 00 00 00 FF FF FF FF | V.2d........????
000EC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000ED0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000EE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000EF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

000F00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
000F40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000F50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000F60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000F70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
000F80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
000FC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000FD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000FE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
000FF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 8
Dumping file SampleVolume, starting at block 8 for 1 block:

001000: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001030: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001040: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001050: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001060: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001070: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001080: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0010C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0010D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0010E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0010F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001130: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001140: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001150: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001160: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001170: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0011C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0011D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0011E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0011F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 9
Dumping file SampleVolume, starting at block 9 for 1 block:

001200: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001230: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001240: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001250: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001260: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001270: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0012C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0012D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0012E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0012F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001300: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001310: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001320: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001330: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001340: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001350: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001360: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001370: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001380: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001390: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0013C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0013D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0013E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0013F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 10
Dumping file SampleVolume, starting at block 10 for 1 block:

001400: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001420: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001430: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001440: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001450: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001460: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001470: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001480: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001490: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0014C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0014D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0014E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0014F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001500: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001510: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001520: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001530: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001540: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001550: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001560: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001570: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001580: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001590: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0015C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0015D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0015E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0015F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 11
Dumping file SampleVolume, starting at block 11 for 1 block:

001600: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001610: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001620: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001630: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001640: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001650: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001660: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001670: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001680: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001690: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0016C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0016D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0016E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0016F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001700: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001710: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001720: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001730: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001740: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001750: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001760: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001770: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001780: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001790: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0017C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0017D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0017E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0017F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 12
Dumping file SampleVolume, starting at block 12 for 1 block:

001800: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001810: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001820: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001830: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001840: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001850: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001860: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001870: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001880: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001890: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0018A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0018B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0018C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0018D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0018E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0018F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001900: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001910: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001920: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001930: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001940: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001950: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001960: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001970: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001980: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001990: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0019A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0019B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0019C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0019D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0019E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0019F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
[student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 13
Dumping file SampleVolume, starting at block 13 for 1 block:

001A00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001A10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001A20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001A30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001A40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001A50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001A60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001A70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001A80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001A90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001AA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001AB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001AC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001AD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001AE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001AF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001B00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001B10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001B20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001B30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001B40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001B50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001B60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001B70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001B80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001B90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001BA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001BB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001BC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001BD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001BE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001BF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$
```

```
[student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 14
Dumping file SampleVolume, starting at block 14 for 1 block:

001C00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001C10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001C20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001C30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001C40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001C50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001C60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001C70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001C80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001C90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001CA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001CB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001CC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001CD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001CE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001CF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001D00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001D10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001D20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001D30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001D40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001D50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001D60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001D70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001D80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001D90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001DA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001DB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001DC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001DD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001DE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001DF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

[student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 15
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 15
Dumping file SampleVolume, starting at block 15 for 1 block:

001E00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001E10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001E20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001E30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001E40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001E50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001E60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001E70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001E80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001E90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001EA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001EB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001EC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001ED0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001EE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001EF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

001F00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001F10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001F20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001F30: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001F40: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001F50: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001F60: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001F70: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
001F80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001F90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001FA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001FB0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
001FC0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001FD0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001FE0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
001FF0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 16
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 16
Dumping file SampleVolume, starting at block 16 for 1 block:

002000: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002030: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002040: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002050: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002060: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002070: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002080: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0020A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0020B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0020C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0020D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0020E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0020F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

002100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002130: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002140: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002150: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002160: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002170: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0021A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0021B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0021C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0021D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0021E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0021F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 17
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 17
Dumping file SampleVolume, starting at block 17 for 1 block:

002200: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002230: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002240: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002250: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002260: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002270: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0022A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0022B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0022C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0022D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0022E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0022F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

002300: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002310: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002320: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002330: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002340: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002350: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002360: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002370: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002380: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002390: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0023A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0023B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0023C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0023D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0023E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0023F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 18
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 18
Dumping file SampleVolume, starting at block 18 for 1 block:

002400: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002420: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002430: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002440: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002450: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002460: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002470: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002480: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002490: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0024A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0024B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0024C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0024D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0024E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0024F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

002500: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002510: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002520: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002530: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002540: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002550: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002560: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002570: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002580: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002590: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0025A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0025B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0025C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0025D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0025E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0025F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 19
```

```
student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 19
Dumping file SampleVolume, starting at block 19 for 1 block:

002600: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002610: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002620: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002630: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002640: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002650: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002660: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002670: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002680: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002690: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0026A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0026B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0026C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0026D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0026E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0026F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

002700: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002710: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002720: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002730: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
002740: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002750: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002760: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
002770: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........
002780: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
002790: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0027A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0027B0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............????
0027C0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0027D0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0027E0: 00 00 00 00 FF FF FF FF  00 00 00 00 FF FF FF FF | ....????....????
0027F0: 00 00 00 00 FF FF FF FF  00 00 00 00 00 00 00 00 | ....????........

student@student-VirtualBox:~/Documents/milestone1test/csc415-filesystem-EssaHusary$ ./Hexdump/hexdump.linux SampleVolume --count 1 --start 20
```

<u>Analysis of the dump:</u>

Within the first block, address 000E00 contains the hexadecimal number 2E which refers to the name of the first directory entry. The value of 2E in decimal is 46, 46 in ascii is “.”. The first entry in our root directory is named “.”. Address 000E10 contains the hexadecimal number 0A. This refers to the size of the “.” directory entry. The decimal equivalent of 0A is 10. In our code the size of the first directory entry was hardcoded to 10. Addresses 000E20-00E2C contains the value 56 1C 32 64 (reversed from little endian). This refers to the created date of the “.” directory entry. The created date represents the number of seconds since January 1, 1979. Which at the time of this screenshot, was 56 1C 32 64 in hex. This equates to decimal as 1444688484 seconds. Addresses 000E30-00E34 contains the value 56 1C 32 64 (reversed from little endian). This refers to the last modified date of the directory entry. This will be equivalent to the created date because the entry has not been modified since it has been created. Addresses 000E3D-000E3F contain the hexadecimal number FF. This refers to the starting block number of our extents node. Since we didn’t need to use extents to initialize this directory we set the starting block number to -1, -1 is equivalent to FFFF in hex. Addresses 000E50–000E54 contain the value 0. This represents the extends count which should be 0 because we have no extends. For every directory entry in our root directory, the same applies, the initial address is the name followed by the size, followed by the created and modified dates, followed the starting block number of our extents and the count of our extents.

Description of our VCB structure:

Our volume control block structure begins at block 0 in the logical block array and contains all of the information necessary to access and control various things in our volume. We have, for instance, storing the size of each block of the volume in a variable called "sizeOfBlocks". This was useful when doing things such as creating the free space map and will be useful in the future when performing allocations using malloc(). We also have the variable "numOfTotalBlocks" which stores the total number of blocks in the partition. It came in handy when creating the free space map and was used in the calculation to determine how many blocks are required for our free space map [(m + (n - 1)) / n]. Another useful piece of information in our VCB is the "magicNumber" variable. This is an essential piece of info as it allows us to determine whether or not our volume has already been formatted. What's more, we have the variables "startingBlockOfFreeSpace", "lengthOfFreeSpaceMap", "startingBlockOfRootDirectory", and "lengthOfRootDirectory", which, as their names suggest, store the starting block number of the free space map, the length of the free space map, the starting block number of the root directory, and the length of the root directory, respectively. Now we can have all kinds of uses for such information, but the most basic uses for these pieces of information involve using them in calls to LBAread() in order to bring them from disk into memory.

Description of our Free Space structure:

Our free space structure is a simple bitmap. We have defined 1 to indicate that a block is free and 0 to indicate that the block is in use. The free space map begins at block number 1 in the logical block array, and in the default, RUNOPTIONS settings in the Makefile where "volumeSize" is 10,000,000 bytes and "blockSize" is 512 bytes, the length of our free space map is 5. So our free space map in this case spans 5 blocks starting from block number 1. Before initializing the bitmap, we had 19,531 bits with a value of 1. Since the volume control block and free space map are now written to disk, the first 6 blocks are 0s because they are now marked used. So the first byte of the free space map looks like this: 0b0000 0011 (or 0x03 in hex).

Description of our Directory System:

…

In our directory system, the first and second directory entry will always be preset and set up to hold the information of the current directory and root directory, respectively. The current

directory's name is presented by ".". Its size is the size of the directory where it is stored. The created and modified time are in seconds representing the time since the Epoch(00:00:00 UTC, January 1, 1970) when the directory entry's information is filled. Also, we are using a structure extentNode which has two field blockNum and count to record the blocks allocated to the directory. The starting block of the current directory entry is the first block where the directory will be stored in the volume. This number is returned from the free space map where the first free block is located. Also, the count of the current directory entry is the number of blocks we need to store the directory. The second directory entry will have the information of the root directory whose name is represented by "..". Its size can be obtained in the volume control block. The created and modified time is set up as the current directory when we first initialize the root directory. All the later creations of a directory can access the root directory and then the second entry to obtain the root's time information. The starting block and length of the root directory can be obtained from the volume control block as well. Both directory entries file types are set as directory.

<u>Table of who worked on which components:</u>

Essa Husary	- Worked with Amit on step 1, formatting the volume. - Worked on the free space map, creating/initializing it (the initialization function is called freeSpaceInit()), and writing the bit manipulation and allocation functions (the allocation function is called getBlocks())
Amit Bal	- Worked on steps, 4a,b,c,d. Created functions to make necessary calculations. For example, writing a function to calculate the total number of blocks needed for the root directory. - Worked on initializing the root directory with Mahdi. Initialized the fields needed for the root directory. As well as created the root directory array. The initialization method is called intiliazeRootDir(). - Worked on writing and reading the root directory from disk using LBARead() and LBAWrite().
Mahdi Hassanpour	- Worked on steps 4e,f , Loop through the directory entries and initialize each to a known free state, then request 6 blocks from the free space system and get the starting block number. And set the 1st entry to ".' and the 2nd entry to ".."

Jinwei Tang	- Work on step 4 g,h, implementing a method to set up the first and second directory entries as current directory entries and root directory entries in every directory

How our team worked together:

Our team tried to meet as often as we possibly could. We met on Discord on Sunday 4/2, Monday 4/3, Wednesday 4/4, and Thursday 4/6 to work on this milestone. We divided the work such that we all had a balanced workload given the amount of understanding each person had with the different parts of the file system and also given the amount of homework people had from other classes. The point was to make things fair for everyone. So we split up the work in the following way. Essa and Amit worked on step 1, sharing ideas with each other, understanding how to check whether or not the volume has been formatted, and figuring out how to initialize the system. Essa worked individually on and concerned himself with the free space system, doing things such as creating/initializing it (the initialization function is called freeSpaceInit()), and writing the allocation and bit manipulation functions (the allocation function is called getBlocks()). Amit worked on steps 4.a - 4.d, performing the necessary calculations to determine how many blocks and entries we need. Mahdi worked on steps 4.a - 4.d, looping through each directory entry structure to be in a known free state. Jinwei worked on steps 4.g and 4.h. While he began his implementation, he gained information about what the directory entry array will look like and also the codes from Mahdi who will implement the precedent procedures. While implementing, he gained updated information on the implementation of the volume control block and free space map through the meetings. He utilized that information to update the way he codes. After he finished the method, he pushed his code to his own branch so others can view them and use the method to coordinate with their sections.

How we approached completing our parts in the assignment:

Essa:

I, Essa, worked on the free space map and worked with Amit on the initFileSystem() function of the fsInit.c file where we formatted the volume and wrote the checks to see whether or not we have already formatted the volume (for subsequent boots of the system). I approached my parts of the assignment in the following way, starting with the formatting of the volume. So my first thought when looking at step 1 of the “Steps for milestone 1” pdf document was, “Okay, so we have to basically allocate some space to store information about our VCB. I can do that. But as for checking to see if the magic numbers match, how can I do that?” So I began thinking about maybe storing the data in a buffer and reading it in somehow. That turned

out to be the wrong approach as I learned in a one-to-one office hours session with you, Professor. I learned that all I had to do was just do an LBAread() starting at block 0 into the space I allocated for the VCB using malloc(). It was that simple. And then I had to check if the field in our VCB for the magic number matches with what was read from disk. Amit thought about it on his own separately, but funny enough, we had the same exact implementation and ideas. Now onto how I approached my individual part, which was the free space map.

As I was reading step 3, I thought about how to initialize the free space map. I first, after reading step 3.a, came up with my own equation to calculate the number of blocks we'll need. But after an office hours session, I realized that the correct equation to use was the equation covered in class, which was [(m + (n - 1)) / n]. I then read step 3.b and did what it said to do which was to malloc the space needed for the free space map. Next, I read 3.c and thought to myself, "Okay, so I remember from one of the in-class lectures that the Professor actually went over how we can manipulate individual bits. Why don't I look at that." And I did. But it was more complex than it seemed and I ran into some issues implementing it (see "Issues and Resolutions" below). But I eventually figured it out. I wrote three functions after I figured out how to do so: setBit(), clearBit(), and initializeMap(). The function setBit() sets the value of a bit to a value of 1, which indicates that a block is free. The function clearBit() sets a bit's value to a value of 0, indicating used. And, initializeMap(), creates the entire map using setBit(), setting every bit in the allocated space to a value of 1 which appears in the hexdump as "FF". So I figured out how to manipulate bits, but I, unfortunately, ran into another issue. I had a hard time writing the allocation procedure, figuring out how to get the caller their desired number of blocks, returning the starting block number, and marking them as used in an efficient way (see "Issues and Resolutions" below). Fortunately, I figured out how to do so and once I did, I proceeded to step 3.d, writing the free space map to disk. And finally, I then proceeded to the final part of that step, step 3.e, and recorded the value of the starting block of the free space map in the volume control block. My part has been completed.

Mahdi:

I, Mahdi. Worked on steps 4e,f , Loop through the directory entries and initialize each to a known free state, then request 6 blocks from the free space system and get the starting block number. And set the 1st entry to ".' and the 2nd entry to "..". The code begins by looping through the array of directory entries and setting each entry's file name, size, date created and modified to 0 or an empty string. This is done to ensure that all entries in the array are properly initialized before being used. Next, the code sets the first two directory entries to "." and "..", respectively. These are typically used to represent the current directory and its parent directory, respectively. The code uses dynamic memory allocation to allocate space for the file names of these two entries since their lengths are different from the other entries. It then copies the file names into the allocated memory and sets the size and date created and modified to 0.

Amit:

I, Amit, worked on making the necessary calculations in order to initialize the root directory, wrote the root directory to disk using LBAWrite(), and loaded the root directory from disk. In order to make the necessary calculations for the root directory I created four different functions. First, I knew I had to calculate the total number of bytes needed for my root directory. I created a function called numBytesForEntries() which multiplied the total number of entries needed by the size of each directory entry. Next, I needed to calculate how many blocks were needed for the root directory. I created a function called numBlocksToAskFreeSpace() which took the total number of bytes needed for the root directory and divided it by the size of each block. Finally, I needed to calculate how many extra entries could be added to the root directory without exceeding the number of blocks allocated for the root directory. I created a function called numExtraEntries(), which modded the total number of bytes for the root directory by the block size. This gave me the total number of extra bytes that could still contain directory entries. I then calculated the total number of extra entries that could be added by dividing the extra bytes by the size of the directory structure. I then created a function called initializeRootDir() which leveraged both Jinwei's and Mahid's code. This function would loop through the total number of directory entries setting each data member to it's proper value. For example the name of the root directory's first entry should be set ".". After initializing the root directory it was time to write it to disk. In order to write the root directory to disk, I needed to use LBAWrite() which takes three parameters, a buffer to write from, a count of blocks to write to, and a starting block number. I got the starting block number by leveraging a method in the free space map called getBlocks() which takes the number of blocks needed for the root directors as a parameter, and returns the starting block number. I then leveraged my numBlocksToAskFreeSpace() function, to get the count of blocks to write to. Finally, I malloced a buffer the size of the root directory. To load the root directory, I had to leverage LBARead() which takes three parameters, a buffer to read from, a count of blocks to read, and a starting block number. I got the starting block number from the VCB structure. I got the count of blocks to read by leveraging my numBlocksToAskFreeSpace() method. Finally, I malloced a buffer the size of the root directory.

Jinwei:

I, Jinwei, worked on setting the root directory's first entry to be "." and second entry to be "..". Since we are doing our part separately for the root directory, I need to figure out what information I can expect to gain from the precedent implementation. In class, we have discussed that "." represents the current directory and ".." represent the root directory. So I need to figure out what information can be stored in their fields. Are those data static or dynamic? First, no matter when we set up the current directory entry for a root directory or normal directory, their names are the same, their sizes are always the size of the directory array where they belong, the time is always the time when we create the instance directory, the starting block and length are always gained from the free space map. So I can set up a method that takes an int startingBlockNum, int size, and directory entry array and sets up the first directory entry "." dynamically. Since the ".." represents the root directory, the setting is static once the root directory is created and written to the volume. Except for the name is always ".." for the root

directory entry, the size, time, or starting block can only be read from the volume control block or from the root directory. Thus, I add another parameter VCB to the parameter list. Also, since this method is just to set up the first two directory entry, the file type of both of them are set to the directory.

Issues and Resolutions:

Essa:

I, Essa, as I said before, worked on the free space map and worked with Amit on the initFileSystem() function of the fsInit.c file where we formatted the volume and wrote the checks to see whether or not we have already formatted the volume (for subsequent boots of the system). I had no issues with completing the latter. However, there were quite a few issues I encountered when completing the former. Thankfully, I had a resolution to every single issue I did have. My first issue was figuring out how to manipulate bits in order to initialize the free space map. My first instinct was to rewatch one of the recorded in-class lectures, as you, Professor, did go over how to do so. That video really helped me understand how the process works, but the implementation details were left to us, the students, to decide. So I experimented, implementing the techniques covered in that lecture, and after all of this experimentation, learning how to shift bits using the bitwise operators and using a whole lot of printfs, I was able to successfully figure out how to manipulate the bits to initialize the map and to write the allocation function. So my resolution was simply rewatching the lecture video and performing lots of testing and abuse printfs.

My second issue revolved around writing the allocation function for the free space. I knew what to do but not how to do it. So I resolved this issue by coming up with a plan. I figured, “Hmm, I can iterate across every byte in the free space map, and in every byte, I can iterate across every bit. I can track how many consecutive 1s (indicating a free block) I encounter, and if we encounter at least one 1, we store the starting block where we found that first 1. If we encounter a 0 before having the number of consecutive 1s equaling the number of blocks the caller wants, we reset the starting block number to 0. However, if the number of consecutive 1s matches the number of blocks the caller wants, we keep that starting block number, clear those bits starting at that starting block number, and return the starting block number of that number of consecutive 1s.” It was a plan that took me a couple of hours to create, but it was well worth it. My resolution was simply to plan the process out on the drawing board.

Mahdi:

I had an error in determining the correct size of the dirEntries array or in the loop bounds for iterating through the array, leading to out-of-bounds access errors. I also encountered errors when allocating memory using malloc(), such as failing to allocate enough memory or allocating memory that is not freed later, leading to memory leaks. encountered errors when initializing the dirEntries array or modifying its contents, leading to data corruption errors or undefined behavior. And the most difficult part of this milestone was combining everybody's code together, for my part, my code needed some variables that were initialed in someone else's code, and making everything compatible was the hardest part. We all created our own separate branches and pushed our code so if there was anything wrong with our code won't break the whole code, and after talking to the professor we realized that wasn't the best thing to do because we were working on the same file, and only 1 person was pushing to the main branch and we were all relying on him. for the next milestone, we are going to create different files instead of different branches, so we can all push to the main branch.

Amit:
While writing the root directory to disk, I had an issue where LBAWrite() would not write the root directory to disk. I couldn't figure out why because I had just malloced my buffer to the size of the root directory which is 13 blocks or 6,656 bytes. The pointer itself was not null, otherwise I would've gotten a segmentation fault. I passed the correct block count and position. I went on a deep dive into the C pointers online and realized that I was passing in the address of the pointer into LBAWrite(). This caused LBAWrite to write to disk a buffer that was only 8 bytes, the size of the pointer instead of the size of the root directory which is 6,656 bytes.

My second issue occurred with storing the results of my methods so that they could exist outside of my file. I went on a deep dive online, and found that a structure would be the best way to store the results of my calculations. I created a struct called directoryEntryCalcs which contains three variables. each of int type. The first variable is called totalBytesForEntries, this variable stores the result of my numBytesForEntries() method. The second variable is called totalBlocks to ask, this variable stores the result of my numBlocksToAskFreeSpace() method. The third and final variable is called totalExtraEntries, this variable stores the result of my numExtraEntries() method. This structure allowed me to reference the results of my calculations anywhere in our program.

Jinwei:
While implementing the method, I have an issue getting the total size of the directory entry array. Usually, we will just use sizeof() to get the length of the malloc length. But after I pass the directory entry array to my method as the argument, the sizeof() always returns 8. So I assume that it is returning the size of the pointer since the array is passed by passing as a pointer to the memory address of the array. I search on Google to see if there are similar occurrences. Then my assumption was proved to be right. Thus, I add another parameter to hold the size of the array so I need not get the size depending on the argument.