

a).

```
-bash-4.1$ gcc LedNumber.c -o LedNumber -g
-bash-4.1$ ./LedNumber
Mapping I/O memory failed - Did you run with 'sudo'?
: Bad file descriptor
```

A file descriptor is an abstract indicator used to access an input/output source such as a file. The program fails during initialization. It tries to open `"/dev/mem"` for reading and writing, but it fails and returns -1 as the file descriptor. Next, the call to `mmap` exits with a perror message and the entire program shuts down.

b).

Initialize(): opens a file descriptor to `"/dev/mem"` with read and write access. It then returns an `mmap` which maps files or devices into memory. It wants to map the size of `gpio_size`, and each address should have an offset of `gpio_address` in the virtual address space. The allocated space should be readable, writable, and shared, and it should be linked to the file descriptor `fd`. The entire initialization returns the address to virtual memory which is mapped to physical or it gives an error.

Finalize(): closes previously open mappings and file descriptors. Removes the mapping previously allocated during initialize. Removes the entirety of it or the size of `gpio_size`. The finalize function then proceeds to close to opened file descriptor.

RegisterRead(): read a 4-byte value from the specified address in the virtual address space. You need to enter the base address returned from `mmap` and then the offset from that address.

RegisterWrite(): write a 4-byte value from the specified address in the virtual address space. You need to enter the base address returned from `mmap` and then the offset from that address. Now, at the correct location, write the value to that address space.

SetLedNumber(): Show the lower 8 bits of integer value on LEDs. This is done by writing to each address in control of the LEDs. The value of an individual bit can be calculated by utilizing modulus and division.