The Buddy System

Write a C program that implements the buddy system on a small memory pool. The program should read a data file formatted like `/u1/junk/cs471/sample.data` The file name should be accepted as a command line argument.

The data file will contain one command per line. The first line will contain the size of the memory pool (in pages). The other lines are of two types.

M: A memory *allocation* request (analogous to a `malloc`). There will be three items on the line: the letter 'M', the process ID, and the number of pages required.

F: A memory *free* request. There will be two items on the line: the letter 'F', and the process ID that is ending.

If there is insufficient memory available, print an error message (error message should go to `stderr`) with the PID, the number of pages requested, and number of pages you tried to allocate (the next power of two); then ignore the process, including any free ('F') requests that appear later. Here is a sample error message.

```
Memory unavailable: 123 100 128
```

When allocating blocks, if there is more than one block of the required size available, use the block with the smallest address. When splitting block, if there is more than one block of the correct size, split the one with the smallest address.

After reading the last line in the input file print the status of your buddy table. Print one line for each block of pages in the table. Each line should show three items: the address of the block (hex), the size of the block (decimal), the process ID if the block is in use or the word 'FREE' if the block is free. Sort the blocks by address, increasing order.

Create a subdirectory of your home directory named `sep04` and save your solution in a file named `buddy.c`. Your solution is due Wednesday September 4, at 3:00PM.