Name: _____

Registration: _____

(10 pts.) Given the following partial module for a 512Kx64 RAM chip, fill in the blanks to complete the module.

```
module RAM (adr, CS, RW, di, do);

     input CS, RW;
     input [18:0] adr;
     input [63:0] di;
     output [63:0] do;
     reg [63:0] d_out;
     reg [127:0] Mem1 [0:524287];

     assign do = (CS && RW)?d_out:64'bz;

     always @(adr or di or CS or RW)
          if (CS && !RW)
               Mem1 [adr] = di;
     always @(adr or CS or RW)
          if (CS && RW)
               d_out = Mem1 [adr];

     initial
          $readmemh ("memory1.dat", Mem1);

endmodule
```

Name: _____

Registration: _____

(10 pts.) Given the following partial module for a 256Kx128 RAM chip, fill in the blanks to complete the module.

```
module RAM (adr, CS, RW, di, do);

     input CS, RW;
     input [17:0] adr;
     input [127:0] di;
     output [127:0] do;
     reg [127:0] d_out;
     reg [127:0] Mem1 [0:262143];

     assign do = (CS && RW)?d_out:128'bz;

     always @(adr or di or CS or RW)
          if (CS && !RW)
               Mem1 [adr] = di;
     always @(adr or CS or RW)
          if (CS && RW)
               d_out = Mem1 [adr];

     initial
          $readmemh ("memory1.dat", Mem1);

endmodule
```