[**You may use one 3.5×5 inch note card and a pencil/eraser or pen.**] Show all of your work clearly in the space provided. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (16 points) Explain the purpose for the following assembler directives:

.cseg —

.dseg —

.byte —

.include —

2. (12 points) Indicate the size of each of the following registers/ports and explain what each is used for:

Stack Pointer —

DDRB —

Program Counter —

3. (12 points) Consider the .lst file on the next page:
(a) How many bytes of memory does the program occupy?

(b) Identify the items in each column below and explain what they represent.

```
Col1   Col2                 Col3  Col4          Col5
00002c ef00                 ldi  temp,0xf0   ; Enable pull-up resistors on PORTC4-7
```

```
AVRASM ver. 2.1.12  C:\Atmel\Projects\exam1.asm Mon Jan 12 04:24:18 2009

C:\Atmel\Projects\exam1.asm(21): Including file 'c:\Atmel\AVRTools\AvrAssembler2\Appnotes\


                         .list

                         .def temp = r16   ; Use R16 as a temp register

                         .cseg             ; Begin code segment
                         .org 0
000000 c029              rjmp init         ; Initialize restart vector

                         .org 0x2a

                 init:
00002a e00f              ldi  temp,0x0f   ; Set keypad cols as output, rows as input
00002b bb04              out  DDRC,temp
00002c ef00              ldi  temp,0xf0   ; Enable pull-up resistors on PORTC4-7
00002d bb05              out  PORTC,temp
00002e ef0f              ldi  temp,0xff   ; Configure PORTB as an output port
00002f bb07              out  DDRB,temp

                 main:
000030 b303              in   temp,PINC   ; Read rows of keypad from PORTC
000031 bb08              out  PORTB,temp  ; Display results on LEDs 5-8
000032 cffd              rjmp main         ; Repeat main

[snip]

ATmega32 memory use summary [bytes]:
Segment   Begin    End      Code   Data   Used    Size   Use%
---------------------------------------------------------------
[.cseg] 0x000000 0x000066     20      0     20   32768   0.1%
[.dseg] 0x000060 0x000060      0      0      0    2048   0.0%
[.eseg] 0x000000 0x000000      0      0      0    1024   0.0%

Assembly complete, 0 errors, 0 warnings
```

4. (6 points) What is the purpose of the stack?

5. (12 points) List at least four instructions that should not be used if the stack pointer has not been initialized.

6. (17 points) Write a subroutine called initPorts that initializes the ports on the ATmega32 microcontroller such that:

- All pins on **PORTB** are configured as input pins.
- All even numbered pins on PORTC are configured as input pins.
- All odd numbered pins on PORTC are configured as output pins.
- PORTD is configured as an output port (all pins).
- Pull-up resistors are enabled for all input pins except for PB7 (the most significant bit of PORTB).

You may assume that the stack pointer has been initialized correctly.

7. (25 points) Write a subroutine called monitorC that continually checks the even pins on PORTC. If all of these pins have the same value, the subrountine should end. Otherwise, these values should be written to the even pins of PORTD. The odd pins on PORTD should be turned off at all times. You should call initPorts (from the previous problem) at the beginning of the subroutine to initialize the ports.