

Consider the .lst file below:

```
                                .cseg
                                .org 0x00
000000 c029                    rjmp init

                                .org 0x2a
                                init:
00002a e008                    ldi r16, 0x08
00002b bf0e                    out SPH, r16
00002c e30f                    ldi r16, 0x3F
00002d bf0d                    out SPL, r16
00002e e005                    ldi r16, 0x03
00002f d000                    rcall recurse

                                recurse:
000030 930f                    push r16
000031 950a                    dec r16
000032 f009                    breq forever
000033 dffc                    rcall recurse

                                forever:
000034 cfff                    rjmp forever
```

Show the contents of the entire stack and the value of stack pointer once the program counter has a value of 0x0034.

Complete the C function below:

```
/**
 * Wait until SW1 on the keypad has been depressed and then return.
 *
 * In order to work properly, the keypad must be connected to PORTB.
 * The function does not assume that PORTB has been configured prior to
 * being called.
 */
void wait_for_sw1()
{
```

```
}
```

(a) (7 points) Describe two differences between C and Java.

(b) (3 points) Write the appropriate javadoc-style documentation for the following function implemented in Lab 3.

```
void lcd_write(uint8_t value, _Bool is_data);
```

Consider the following code:

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "adc.h"

int main()
{
    sei();
    init_adc(); // Assume this is implemented elsewhere
    DDRB = 0xff
    while(1);
    return 0;
}
```

Explain in detail what the addition of the following lines would change in the machine code generated by the compiler.

```
ISR(ADC_vect)
{
    PORTB = ADCH;
}
```

Implement the following function in assembly:

```
/**
 * Silly function to test students' ability to mix C and assembly code.
 *
 * @param for_ddrb Value to be sent to DDRB.
 * @param for_portb Value to be sent to PORTB.
 * @return a 16-bit value with for_ddrb in the upper byte
 *         and for_portb in the lower byte.
 */
extern uint16_t quiz5(uint8_t for_ddrb, uint8_t for_portb);
```

The .s file has been started for you:

```
#define __SFR_OFFSET 0
#include <avr/io.h>

.global quiz5

.section .text
```

Suppose that an array of `uint8_t`s exists and that there is a pointer that points to an arbitrary element in the array. Implement a function that is passed a pointer to an `uint8_t`. The function should return an `uint16_t` that is the sum of all the all the positive integer values between the value currently pointed to and the next element in the array that has a value of 0. For example, if we had:

```
-----  
|  3 |  0 | 15 | 244 | 13 |  1 |  0 | 255 | 112 |  0 |  
-----  
          ^
```

where the pointer passed to the function is point to the 15, then the function should return 273 (= 15 + 244 + 13 + 1).



Implement a function called `uart_putc` that will take one `unsigned char` as an argument and send it through the USART subsystem to a device connect via the serial cable. In addition to implementing the function, describe each part of your implementation in English. Your description should be precise and detailed.

Consider the `rectangle.h` file developed in lecture (on back of this page).

(a) Write the line(s) of code necessary to create a `Rectangle` with one corner at (100,100) and the other at (200,200).

(b) What does the `static` keyword in front of the last function prototype do?

(c) Why is the `static` keyword placed in front of the last function prototype?


```
typedef struct {
    int x;
    int y;
} Point;

typedef struct {
    Point ul;
    Point lr;

    int (*getArea)(const struct Rectangle const *this);
    int (*getPerim)(const struct Rectangle const *this);
    void (*moveTo)(struct Rectangle *this, const Point to);
} Rectangle;

void _Rectangle(Rectangle *this, int ulx, int uly, int lrx, int lry);

static int getArea(const Rectangle const *this);

static int getPerim(const Rectangle const *this);

static void moveTo(Rectangle *this, const Point to);
```