16/16

1 **Glossary Items**.
Define these terms in ways that distinguish them from each other.

2/2 a.) Parallel port.
A parallel port is a set of I/O pins that can be configured in various ways (input, output, pull-up resistors, etc.) and driven in a word-wide manner.

b.) Serial port
An individual pin used as an input or as an output. It sends or received one bit at a time. A word of data has all the individual bits shifted out of it one bit at a time, each bit is sent separately, and then the bits are shifted into a register at the other end of the transmission in order to re-create the word.

2 **Parallel Ports on the Arduino**.

2/2 In section 1.1of the ATmega328P datasheet, the available parallel ports on this microcontroller are named. What are the names of the parallel ports on the ATmega328P? (Hint: There are three.)
Port B, Port C, and Port D.

3 **Parallel Ports in the Arduino IDE.**

2/2 Procedures provided by an IDE can cast a new perspective over the programmer's view of the underlying hardware. Arduino IDE makes it appear as if the Arduino Uno has one 13-bit-wide parallel port. It furthermore encourages each individual bit to be configured individually such that one could alternatively think of the Arduino Uno as having 13 parallel ports, each just 1 bit wide. From the documentation of the Arduino IDE, name the procedures available for manipulating the parallel port pins. (Hints: Within the Arduino documentation the concept of a parallel port is often described using the initialism GPIO or the phrase, *digital pins*. There are ~~four~~ three procedures provided for manipulating these pins, three of which are used in the "Blink" example.)
The procedures are `pinmode()` and `digitalRead()` and `digitalWrite()`

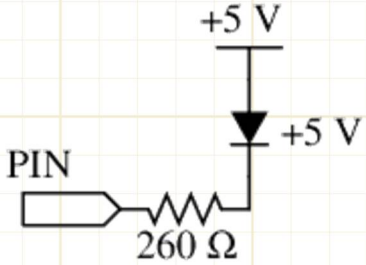
4 **Light an LED.**

4/4 a.) What current is correct for a Light-On (brand) LTL-4234 (Manf. p/n) green LED?
The normal or "Test Current" is 10 mA

b.) What voltage across the LED will result if it is properly driven with the correct current?
From the datasheet, page 4, the table near the left side of the page, about half-way down the page, for 10 mA forward current the voltage drop across the LED will be 1.96 V or about 2.0 V.

c.) An Arduino Uno operates on a 5 V power supply.
An output pin, when driven low under load will typically be at about 0.4 V.
What resistance in series with the LED is needed to establish the correct operating current in the LED?
Voltage drop across the resistor will be $(5\text{ V}) - V_D - V_{OL} = (5\text{ V}) - (0.4\text{ V}) - (2.0\text{ V}) = 2.6\text{ V}$
Then $R = (2.6\text{ V})/(10\text{ mA}) = 0.26\text{ k}\Omega = 260\ \Omega$

d.) Draw a schematic showing how to connect the LED so that it lights up when the pin is at logic-0. (LOW)



5 **Read about multiplexing and charlieplexing LEDs..**

6/6

Suppose a clock display is needed. Seven-segment displays will be used. Four digits are needed, two for the hour display, and two for the minute display. The seven-segment displays are available as common-cathode or common anode—they look identical but are electrically connected differently. Note that all of one digit's display must be common cathode or all of it must be common anode. Within a digit you may not mix common cathode and common anode. (Or else it is not "common"!)

a.) Draw a schematic to show how an Arduino Uno can drive such a four-digit display. Take advantage of the human eye's persistence of vision. (All segments and digits do not need to be simultaneously on so long as the period at which any segment flickers is less than about 15 ms.)

A schematic is shown below. Transistors Q1 thrugh Q4 may be type TN0606 or equivalent which are n-channel MOSFETs. (Supertex is one brand that makes these.)
The 7-segment displays are common cathode types.
The NMOS transistors may be replaced with NPN bipolar transistors. In this case series resistors of about 330 Ω will be needed in series with the pins to regulate the base drive currents.
Typically the decimal point is not used for a clock display so its wiring may be omitted, saving a pin.

b.) Explain how to make your display show 12:34. Give some detail specifically on how to make the "1" in the display show up properly. Then generalize from that to show how to make the other digits display.

First do whatever computing is needed to compute the time. I assume this will take less than 1 ms. Then display the ten's digit of the hour and delay a few milliseconds (about 4 ms will do). Then display the one's digit of the hour and delay, then the ten's digit of the minute and delay and then the one's digit of the minute and delay. Then loop back to recompute the time. Repeat this loop forever.

All the pins shown should be set up as outputs. To turn off all the digits, drive D2 through D5 LOW (turns Q1 through Q4 off) and drive D4 through D11 LOW (removes voltage source from anodes).

To display a particular number, drive D6 through D13 with the font information. Specifically to display "1" in the ten's place of the hour's digits, drive D5 HIGH and D4 through D2 LOW (turns on Q4 and turns of Q3, Q2, and Q1). Also segments b and c should illuminate, thus drive D7 and D8 HIGH while driving D6, and D9 through D13 LOW.

After displaying the hour's ten's digit for a few milliseconds turn it off (D5 LOW) and turn on the next digit (D4 HIGH) and drive the correct font information onto D6 through D13, etc.