### Question 1.
A) B) C) A B or C
### Question 2.
A) nonintrusive B) minimally intrusive
A B or C
### Question 3.
A) nonintrusive B) minimally intrusive
A B or C
### Question 4.
A) nonintrusive B) minimally intrusive
A B or C
### Question 5.
A) nonintrusive B) minimally intrusive
A B or C
### Question 6.
A) long B) high C) highly intrusive D) stack RAM
A B or C
### Question 7.
Yes/no (why)
no It is atomic with interrupts disabled.
### Question 8.
A) EEPROM B) global RAM
A B C or D
### Question 9.
A) EEPROM B) global RAM
A B C or D
### Question 10.
A) EEPROM B) global RAM
A B C or D
### Question 11.
A) EEPROM B) global RAM
A B C or D
### Question 12.
A) EEPROM B) global RAM
A B C or D
### Question 13.
G) TCNT equals TC7
A B C or D
### Question 14.
The 1mA LED current means you can connect it directly to the 9S12C32.
\[ R = \frac{(5-2-0.8V)}{1mA} = 2.2V/1mA = 2200 \, \Omega \]

### Question 15.
You need an interface that can sink 50 mA, like the 2N2222. Since it is a +5V stepper, we will connect one side of the stepper coil to +5V. Since the stepper coil needs 50 mA, and \( h_{fe} \) is 100, the base current needs to be at least 0.5mA. When the 9S12C32 outputs a high, the voltage on PT7 will be at least 4.2V (\( V_{OH} \)). The 2N2222 will be active if \( V_{be} \) is over 0.6V. To select \( R \) for this interface, we set \( \frac{(4.2V-0.6V)}{R} > 0.5mA \). \( R < \frac{(4.2V-0.6V)}{0.5mA} = 7200 \, \Omega \).

### Question 16.
StateType fsm[2]=
\[
\{(300,\{2,4,6,8\}),\{S1,S0,S1,S0\}),
\{(500,\{0,5,10,15\}),\{S1,S1,S0,S1\}) \}
\]

```c
void InitFSM(void){
    TSCR1 = 0x80; // enable TCNT
    TSCR2 = 2; // divide by 4, creating 1 MHz TCNT
    TIOS |= 0x04; // channel 2 is output compare
    TIE |= 0x04; // arm output compare 2
    DDRM = 0x3C; // PM5,4,3,2 outputs PM1,0 inputs
    Pt = S0; // initial state
    TC2 = TC2+300; // wait for S0
    asm cli // enable
}

void interrupt 10 OC2han(void){unsigned char input;
    TFLG1 = 0x04; // acknowledge
    input = PTM&0x03; // PM1 and PM0 are inputs
    PTM = 4*(Pt->Out[input]); // output depends on input and state
    Pt = Pt->Next[input]); // next depends on input and state
    TC2 = TC2+Pt->Time;
}
```