Question **1**

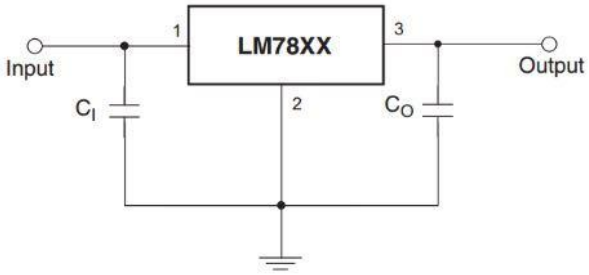
Incorrect

Mark 0.00 out of 1.00

Flag question

According to the datasheet of the LM7805 linear regulator, in a typical fixed output voltage application, which capacitor is required for improved stability and transient response?

Volgens die datavel van die LM7805 linéere reguleerder, in 'n tipiese konstante uittree spannningskonfigurasie, watter kapasitor word vereis vir beter stabiliteit en oorgangsgedrag?



Select one:

- a. CO = 22uF
- b. CO = 100nF
- c. CO = 10uF
- d. CO = 330nF
- e. CI = 100nF
- f. CI = 330nF ✗
- g. CI = 22uF
- h. CI = 10uF

Your answer is incorrect.

The correct answer is: CO = 100nF

Question **2**

Partially correct

Mark 0.50 out of 1.00

Flag question

Two devices that communication with each other using UART are powered independently. It is necessary to include a series resistor on the communications lines because...

Twee toestelle wat met mekaar kommunikeer deur 'n UART kanaal word onafhanklik van krag voorsien. Dit is noodsaaklik om 'n serie weerstand op die kommunikasie lyne te hê omdat...

Select one or more:

- [x] a. one or both devices may be damaged due to parasitic current.
  een of beide toestelle kan beskadig word as gevolg van parasitiese stroom. ✓
- [ ] b. it results in improved signal quality.
  dit veroorsaak verbeterde sein kwaliteit.
- [ ] c. it translates voltage levels from one device to the other.
  dit transleer spanningsvlakke van een toestel na die ander.
- [ ] d. it pulls up the line to Vcc in case the transmitting device does not have internal pull-up resistors.
  dit die lyn op trek na Vcc ingeval die sender toestel nie interne optrekweerstande het nie.
- [ ] e. if one device is on and the other off, current may flow from the powered device to the other, causing it to switch on (with unexpected behaviour).
  as een toestel aan is en die ander af, kan stroom vloei van die aangedrewe toestel na die ander, wat veroorsaak dat dit aanskakel (met onverwagte gedrag).

Your answer is partially correct.

You have correctly selected 1.
The correct answers are: one or both devices may be damaged due to parasitic current.
een of beide toestelle kan beskadig word as gevolg van parasitiese stroom., if one device is on and the other off, current may flow from the powered device to the other, causing it to switch on (with unexpected behaviour).
as een toestel aan is en die ander af, kan stroom vloei van die aangedrewe toestel na die ander, wat veroorsaak dat dit aanskakel (met onverwagte gedrag).

Question **3**

Incorrect

Mark 0.00 out of 1.00

Flag question

Consider the code below, of a main loop and UART interrupt service routine (ISR). Select the missing keyword <???> from the drop-down list

In die onderstaande bronkode van 'n voorbeeld hooflus en onderbrekings-diens-roetine (ISR), kies die regte sleutelwoord vir <???> van die lys.

```
<???> uint8_t rx_data;
<???> uint8_t uart_byte_received;

void r_uart1_interrupt_receive(void)
{
   rx_data = RXD1;
   uart_byte_received = 1;
}

void main
{
   while (!uartbyte_received)
   { }  // wait until a character arrives

   printf("%d", rx_data);
}
```

Select one:

- a. const
- b. volatile
- c. #define ✗

Your answer is incorrect.

The correct answer is: volatile

Answer: (penalty regime: 0,25,50,75,100 %)

```
1  uint8_t CheckBuffer(char* buf, uint8_t len)
2
3  void main (void)
4
5  void finction void
6      if(buf[0] =='$' && buf[1] == 'a')
7      {
8          len[0] = '$';
9          len[0] = '0';
10     }
11     else if (buf[0] == '$' && buf[1] == 'b')
12     {
13         len[0] = '$';
14         len[0] = '0';
15     }
```

Syntax Error(s)

```
prog.c: In function 'CheckBuffer':
prog.c:14:1: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'void'
 void finction void
 ^
prog.c:20:5: error: expected declaration specifiers before 'else'
     else if (buf[0] == '$' && buf[1] == 'b')
     ^
prog.c:26:12: error: expected '=', ',', ';', 'asm' or '__attribute__' before '{' token
 int main() {
            ^
prog.c:29:1: error: expected '{' at end of input
 }
 ^
```

Question author's solution:

```
uint8_t CheckBuffer(char* buf, uint8_t len)
{
    uint8_t paramlen = len-4;

    if ((len < 3) || (buf[0] != '$') || (buf[len-2] != '\r') || (buf[len-1] != '\n'))
        return 3;
    else if (buf[1] == 'a')
    {
        if (paramlen == 0)
            return 0; // ok!
        else
            return 2; // incorrect param length
    }
    else if (buf[1] == 'b')
    {
        if (paramlen == 1)
            return 0; // ok!
        else
            return 2; // incorrect param length
    }
    else
        return 1;
}
```

Incorrect

Marks for this submission: 0.00/4.00.