What is the effect of executing the following code on a PIC16877A microcontroller?

banksel TRISB

movlw B'00100001'

movwf PORTB

comf   TRISB

- ○ The value B'11011110' appears on PORTB
- ◉ The value B'00100001' appears on PORTB   ✘
- ○ RB0 and RB5 are configured as output
- ○ RB0 and RB5 are configured as input
- ○ None of the other choices

The correct answer is:
RB0 and RB5 are configured as output

Consider the circuit below. To display number 3 on the 7-segment display, assuming it is common cathode, we need to execute the following
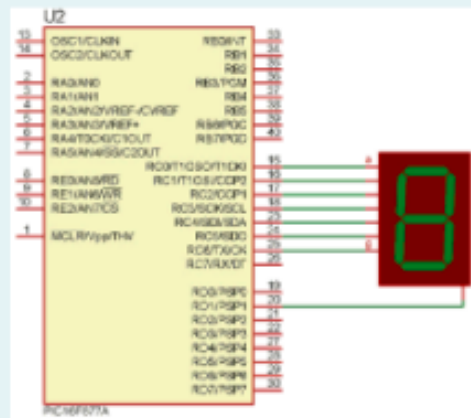
○ TRISC = 0;

TRISD = 0;

PORTC = 0b10110000

PORTD = 0

○ None of the other choices

○ TRISC = 0;

TRISD = 0;

PORTC = 0b01001111

PORTD = 0

○ TRISC = 0;

TRISD = 0xFF;

PORTC = 0b01001111

PORTD = 0

○ TRISC = 0;

TRISD = 0;

PORTC = 0b01001111

PORTD = FF

The correct answer is:
TRISC = 0;
TRISD = 0;
PORTC = 0b01001111
PORTD = 0

Assuming that the SEND_CHAR send SEND_CMD subroutines are already available, the character that is stored in the CGRAM when executing the following code on a PIC16F877A connected to an LCD *contains*

```
        MOVLW   0X68
        CALL    send_cmd
        CALL    SUB1
        MOVLW   0
        CALL    send_char
        MOVLW   0
        CALL    send_char
        CALL    SUB1
SUB1    MOVLW       3
        MOVWF       CNT
L       MOVLW       0X1F
        CALL        send_char
        DECFSZ      CNT, F
        GOTO L
        RETURN
```

○ One dark rectangle in the middle of the character

○ Three dark rectangles on top of each other

○ None of the other choices

◉ Two dark rectangles on top of each other   ✔

○ Two dark rectangles beside each other

The approximate delay generated when the following code is executed on a PIC16F877A microcontroller that is running at 2 MHz is

```
char CNT ;
CNT = 10 ;
while (CNT < 21)
{
    _delay(110) ;
    If (CNT== 15 || CNT == 18) {
            _delay(50) ;
    }
    CNT++;
}
```

○ 2.52 msec   ✗

○ 0.63 msec

○ 5.44 msec

○ 0.66 msec

○ None of the given numbers

○ 2.62 msec

○ 1.36 msec

The correct answer is:
2.62 msec

For the following code, what is the logic operation performed on the values in locations X and Y to produce value stored in location RES?

```
COMF        X, W
IORWF       Y, W
MOVWF       TEMP
COMF        TEMP, W
ANDWF       X, W
MOVWF       RES
```

- ◉ RES = (X'+Y)' X
- ○ RES = (X'Y)' + Y
- ○ None of the other choices
- ○ RES = (X'Y')' + Y
- ○ RES = (X'+Y')' X

✔

The correct answer is:
RES = (X'+Y)' X

For the following code, what is the arithmetic operation performed on the values in locations X, Y and Z to produce value stored in location RES?

```
        MOVLW       D'12'
LX      ADDWF       X, W
        DECFSZ      Y, F
        GOTO        LX
        ADDWF       Z, W
        MOVWF       RES
```

- ○ None of the other choices
- ○ RES = (X+12)*Z + 12 + Y
- ○ RES = X*Z + 12 + Z
- ○ RES = (X+12)*Y + 12 + Z
- ○ RES = X*Y + 12 + Z

The correct answer is:
RES = X*Y + 12 + Z

Which of the following statements is correct?

○ Writing three EQU statements in your programs consumes three memory location in the program memory ✗

○ All other statements are wrong

○ You may need to perform context saving when you use macros

○ Modular programming is used in case we need to divide the program into dependent tasks only.

○ The use of the ORG directive is optional with subroutines to specify the address of the subroutine

The correct answer is:
The use of the ORG directive is optional with subroutines to specify the address of the subroutine

Which of the following statements is correct?

- ◉ Pull-down resistors make sure that the voltage on the wire is low ✔
- ○ SPST switches can be used to supply some instrument with two different voltages
- ○ Regulators can be used to generate clock signals with precise frequency
- ○ A simple inverter can be used for switch de-bouncing
- ○ Resonators can be used maintain a constant voltage level

The correct answer is:
Pull-down resistors make sure that the voltage on the wire is low

Which of the following statements is **correct** regarding MPLAB

○ Labels in MPLAB are not case-sensitive

◉ Instructions are not case-sensitive ✔

○ Writing a decimal constant with MOVLW const instruction does not require base specifier

○ The binary numbering system is not available in the Watch tool in MPLAB

○ All other statements are wrong

○ Only programs that execute sequentially need the END directive

The correct answer is:
Instructions are not case-sensitive

If the following code is executed while an LCD is connected to the microcontroller, then which of the following statements is true?

```
MOVLW  0x70
CALL    send_cmd
```

- ⦿ Selects location number 6 in CGRAM ✔
- ○ Selects location number 3 in CGRAM
- ○ None of the other choices
- ○ Selects location number 4 in CGRAM
- ○ Selects location number 5 in CGRAM

The correct answer is:
Selects location number 6 in CGRAM

Drag and drop the correct instructions to complete the assembly program that corresponds to the following C code.

```c
char CNT, X;
X = X << 1 ;
CNT = X + 3 ;
for (char Y = 0; Y < CNT; Y=Y+2)
{
        X = X + CNT ;
}
```

===

```
        BCF         STATUS, C    ✔
        RLF         X, F         ✔
        MOVLW       0X03
        ADDWF       X, W         ✔
        MOVWF       CNT          ✔
        CLRF        Y
LLL     BTFSS       STATUS, C    ✘
        ADDWF       X, F
        INCF        Y, F         ✔
        INCF        Y, F
        MOVF        CNT, W
        BCF         STATUS, Z    ✘
        MOVF        CNT, W       ✘
        GOTO        LLL
```

===

```
        SUBWF    Y, W          BTFSC      STATUS, C
```

The correct answer is:

Drag and drop the correct instructions to complete the assembly program that corresponds to the following C code.

```c
 char CNT, X;
X = X << 1 ;
CNT = X + 3 ;
for (char Y = 0; Y < CNT; Y=Y+2)
{
        X = X + CNT ;
}
```

```
=========================================
        [BCF STATUS, C]
        [RLF X, F]
        MOVLW          0X03
        [ADDWF X, W]
        [MOVWF CNT]
        CLRF            Y
LLL     [MOVF CNT, W]
        ADDWF           X, F
        [INCF Y, F]
        INCF            Y, F
        MOVF            CNT, W
        [SUBWF Y, W]
        [BTFSS STATUS, C]
        GOTO            LLL


=========================================
```