Summative assessment

Task 1 .

1. Name **one output device** that is built into the micro:bit.

| **Ans:** |  |
| --- | --- |

1. Name **at least two input devices** that are built into the micro:bit.

| **Ans:** |  |
| --- | --- |

1. You would like to build a micro:bit project that provides **audio** feedback to the user, e.g. a beep or a short tune. Which of the following sentences is true?
   1. It is not possible to generate audio output with the micro:bit.
   2. It is possible to generate audio output with the micro:bit, by connecting speakers to the built-in audio port.
   3. It is possible to generate audio output with the micro:bit, by connecting speakers to the GPIO pins.

| **Ans:** |  |
| --- | --- |

1. Which **sensor**, built into the micro:bit, would be used in order to detect ‘gestures’, such as shaking or titling?

A. Radio antenna

B. Accelerometer

C. Magnetometer

| **Ans:** |  |
| --- | --- |

Task 2 .

|  |  |  |
| --- | --- | --- |
|  | Figure 1: Control the LED |  |

1. In Figure 1, the LED is supposed to be controlled (turned on or off) using pin 0. The arrow shows that one of the crocodile clips has not been connected to a pin. Which pin should it be connected to?
   1. Power (3V)
   2. Ground (GND)

| **Ans:** |  |
| --- | --- |

1. In Figure 1, you are supposed to use pin 0 to control the LED (turn it on or off). Which piece of code would you use to turn the LED on, assuming the wiring is correct?
   1. pin0 = 1
   2. pin0.read\_digital() == 1
   3. pin0.write\_digital(1)

| **Ans:** |  |
| --- | --- |

Task 3 .

|  |  |  |
| --- | --- | --- |
|  | Figure 2: Detect a signal. |  |

1. In Figure 2, pin 0 is to be used to detect if a connection is made between the two jumper wires. The arrow indicates that one of the crocodile clips has not been connected to a pin. Which pin should it be connected to?
   1. Power (3V)
   2. Ground (GND)

| **Ans:** |  |
| --- | --- |

1. In Figure 2, pin 0 is to be used to detect if a connection is made between the two jumper wires. Which piece of code would you use to check if the wires are connected, assuming the wiring is correct?
   1. pin0 == 1
   2. pin0.read\_digital() == 1
   3. pin0.write\_digital(1)
   4. pin0.is\_touched()

| **Ans:** |  |
| --- | --- |

Task 4 .

| Program 1 | |
| --- | --- |
| 1  2  3  4  5  6  7 | from microbit import \*  state = 0  while True:  if button\_a.was\_pressed():  state = 1 - state  display.show(state)  display.show(">") |

1. When Program 1 is executed, what will the user see on the micro:bit display after button A has been pressed **once**?

| **Ans:** |  |
| --- | --- |

1. When Program 1 is executed, what will the user see on the micro:bit display after button A has been pressed **twice**?

| **Ans:** |  |
| --- | --- |

1. When will line 7 of Program 1 be executed?
   1. Line 7 will never be executed.
   2. Line 7 will be executed once, after button A has been pressed twice.
   3. Line 7 will be executed every time button A is pressed.
   4. Line 7 will be executed constantly, i.e. in every round of the while loop.

| **Ans:** |  |
| --- | --- |

Task 5 .

| Program 2 | |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | from microbit import \*  import radio  radio.on()  start = running\_time()  message = radio.receive()  while message != "ping":  message = radio.receive()  end = running\_time()  elapsed = . |

1. **Fill in the gap** in line 9 of Program 2 above, so that the value of the elapsed variable is the (approximate) number of **seconds** between the time that line 4 was executed and the time that line 8 was executed.

**Note**: The running\_time function returns the number of milliseconds (thousandths of a second) since the micro:bit was switched on or restarted.

| **Ans:** |  |
| --- | --- |

1. **Describe in one short sentence** the purpose of lines 5 to 7 in Program 2 above.

| **Ans:** |  |
| --- | --- |

Task 6 .

You are asked to create a program for the micro:bit that:

| * Starts by lighting up a single pixel in the top left-hand corner of the 5⨉5 LED matrix. |  |  |  |
| --- | --- | --- | --- |
| * Whenever button B is pressed, the pixel that is currently lit up is turned off, and the one to its right is lit up. |  |  |  |
| * If the pixel that is currently lit up is already at the end of the current row, then the next pixel to light up should be the first pixel in the next row. |  |  |  |

These are the program statements you can use, provided **in no particular order**.

|  | from microbit import \*  display.set\_pixel(x, y, 0)  display.set\_pixel(x, y, 9)  if button\_b.was\_pressed():  if x > 4:  x = 0  x = x + 1  y = 0  y = y + 1  while True: |
| --- | --- |

**Note**: The set\_pixel(*x*,*y*,*b*) method of the display sets the pixel at coordinates *x*, *y* to brightness *b*. The *x* and *y* coordinates range from 0 to 4 (inclusive), starting from the top left-hand corner. The brightness *b* ranges from 0 to 9 (inclusive).

Rearrange the statements and use indentation where necessary, so that the resulting program works as specified. You can use statements more than once.

| **Ans:** | 1  2  3  4  5  6  7  8  9  10  11 |  |
| --- | --- | --- |

1. Select one of the statements below to **replace** the while True statement, so that the program terminates when button B is pressed and the currently illuminated pixel is at the bottom right-hand corner of the LED matrix.
   1. while x <= 4:
   2. while y <= 4:
   3. while not True:

| **Ans:** |  |
| --- | --- |

Resources are updated regularly - the latest version is available at: [the-cc.io/curriculum](http://the-cc.io/curriculum).



This resource is licensed by the [Raspberry Pi Foundation](https://www.raspberrypi.org/) under a Creative Commons At

tribution-NonCommercial-ShareAlike 4.0 International license. To view a copy of this license, visit, see [creativecommons.org/licenses/by-nc-sa/4.0/](https://creativecommons.org/licenses/by-nc-sa/4.0/).