



This program uses two 7 segment displays to show numbers up to ninety nine. Each digit is "unlocked" by pulling the signal "LOW", and then the relevant numeral is sent to it. The maths works out what the "tens" digit and the "ones" digit needs to display.

Use the subroutines as made in "BCD to 7 segment 1", and add a subroutine called "sendNum" as shown.



```

Commands
ZERO
digitalWrite() # bit1 LOW
digitalWrite() # bit2 LOW
digitalWrite() # bit3 LOW
digitalWrite() # bit4 LOW

Commands
ONE
digitalWrite() # bit1 HIGH
digitalWrite() # bit2 LOW
digitalWrite() # bit3 LOW
digitalWrite() # bit4 LOW
    
```

```

Commands
test thisNum = 0
if then ZERO

test thisNum = 1
if then ONE

test thisNum = 2
if then TWO

test thisNum = 3
if then THREE

test thisNum = 4
if then FOUR

test thisNum = 5
if then FIVE

test thisNum = 6
if then SIX

test thisNum = 7
if then SEVEN

test thisNum = 8
if then EIGHT

test thisNum = 9
if then NINE

sendNum
    
```

etc. up to number NINE



```
setup
  int = variable bit1 value 4
  int = variable bit2 value 5
  int = variable bit3 value 6
  int = variable bit4 value 7
  int = variable button value 2
  int = variable ones value 10
  int = variable tens value 11
  int = variable numCount value 0

loop
  int = variable tenCount value numCount ÷ 10
  int = variable oneCount value numCount - tenCount × 10
  int = variable thisNum value tenCount
  digitalWrite() # tens LOW
  sendNum
  digitalWrite() # tens HIGH
  int = variable thisNum value oneCount
  digitalWrite() # ones LOW
  sendNum
  digitalWrite() # ones HIGH
  delay ms milliseconds 500
  test digitalRead() # button == LOW
  while Commands delay ms milliseconds 10
  int = variable numCount value numCount + 1
  test numCount == 100
  if then
    int = variable numCount value 0
```