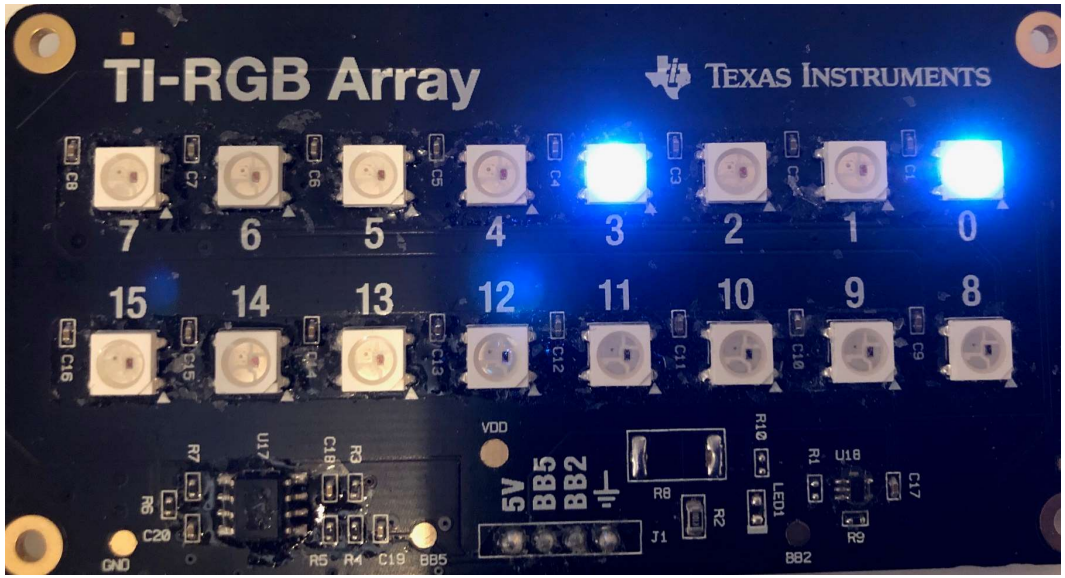
Binary Trainer using RGB Array

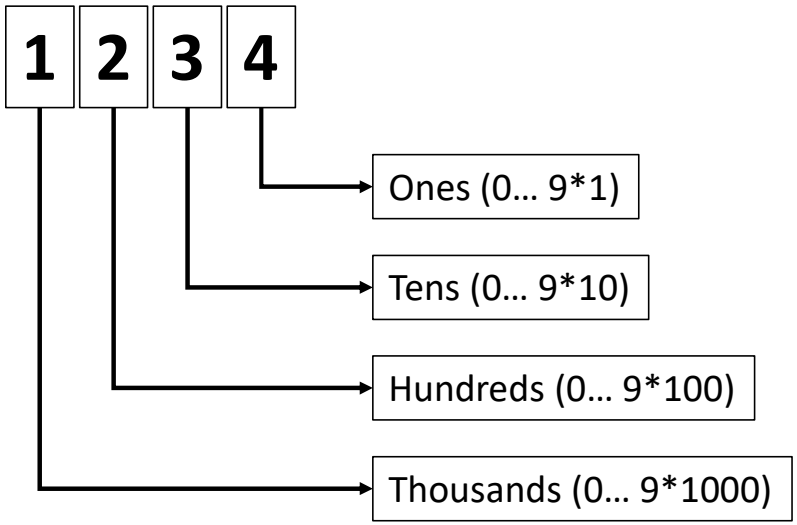


Which binary number is shown here?

HMH 12/10/19

Binary Trainer using RGB Array

Number systems – Humans use a decimal number system (0…9)



4*1 = 4

3*10 = 30

2*100 = 200

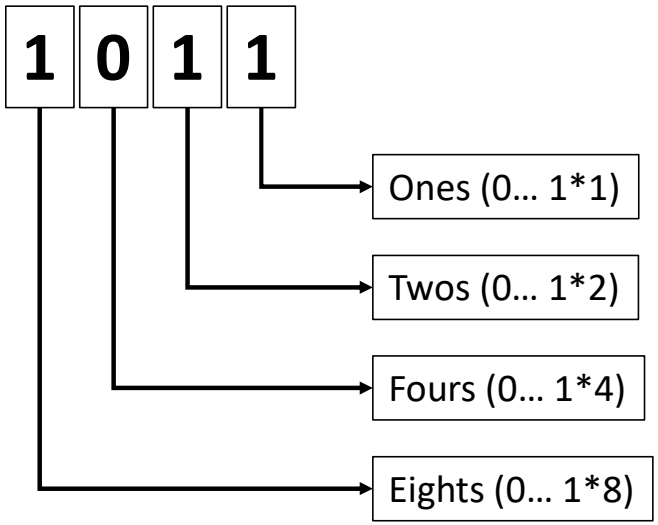
1*1000 = 1000

Sum = 1234

HMH 12/10/19

Binary Trainer using RGB Array

Computers use a binary number system (0…1)



1*1 = 1

1*2 = 2

0*4 = 0

1*8 = 8

Sum = 11 (Dez)

HMH 12/10/19

Binary Trainer using RGB Array

The main program:

```
Define bintst()=                          :  Main program (uses subroutines)
Prgm                                      :
:Local w,r,z,a,x,gd,bd                    : All main prog variables are kept local
:gd:=0                                    : Counter variable for correct answers
:bd:=0                                    : Counter variable for wrong answers
:Send "CONNECT RGB "                      : Connect RGB Array with Hub
:w:=0                                     : Initialize bit-width to allow at least one run thru <while> loop
:While w<1 or w>8                         : Any bit-width between 2 bits and 8 bits is ok
:  Request "Bit-width (2-8): ",w          :
:EndWhile                                 : Keep asking as long as garbage is entered
:z:=2^(w)-1                               : Calc largest decimal number based on bit-width
:Disp "largest decimal number: ",z        :
:x:=1                                     : Initialize loop vars to allow at least one run thru <while> loop
:a:=1                                     :
```

HMH 12/10/19

Binary Trainer using RGB Array

Main prog con't:

```
:While x>0 and a>0
:  r:=randInt(1,z)
:  dispbin(r,"w")
:  Request "Which decimal number do you
see? (stop=cancel)",a,0,x
:  If x>0 and a>0 Then
:      If a=r Then
:        green(a)
:        gd:=gd+1
:      Else
:        red(a,r)
:        bd:=bd+1
:      EndIf
:    Wait 1
:  EndIf
:EndWhile
```

```
:x will be 1 if program is canceled, a is user input, must be > 0
:Create random number
:display binary number on RGB Array (subroutine)
:Ask user to enter number shown on RGB Array in decimal
:
:As long as pgm is not cancelled and answer is a valid number
:Number entered is equal to random number?
: Yes: flash correct number in green on RGB Array (subroutine)
: Increment counter of correct answers
:
:No: Alternate flash correct and incorrect number on RGB Array
:Increment counter of incorrect answers
:
: Pause for a second
:
:
```

HMH 12/10/19

Binary Trainer using RGB Array

Main prog cont'd:

```
:Send "SET RGB all 0 0 0"
:DispAt 6,"Numbers in total: ",gd+bd
:DispAt 7,"Correct answers: ",gd
:DispAt 8,"Wrong answers: ",bd
:EndPrgm
```

```
:Pgm is cancelled: turn off all RGB LEDs
:Show stats and…
:
:
: end the program!
```

HMH 12/10/19

Binary Trainer using RGB Array

<Dispbin> subroutine:

```
: Define dispbin(d,rgbstr)=                  : d holds the number to be displayed, rgbstr selects the LED color
Prgm                                         :
:Local i,z,r,g,b                             : all variables are kept local
:Send "set rgb all 0 0 0"                    : First, turn off any LEDs left on from previous call
:If rgbstr="w" Then                          : Set RGB value for white LED
:  r:=255                                    :
:  g:=255                                    :
:  b:=255                                    :
:ElseIf rgbstr="g" Then                      : Set RGB value for green LED
:  r:=0                                      :
:  g:=255                                    :
:  b:=0                                      :
:ElseIf rgbstr="r" Then                      : Set RGB value for red LED
:  r:=255                                    :
:  g:=0                                      :
:  b:=0                                      :
:EndIf                                       :
```

Binary Trainer using RGB Array

<dispbin> cont'd:

```
:For i,8,1,−1                                                  : Start with MSB, work towards LSB
:  z:=int(((d)/(2^(i-1))))                                     : Break decimal number into powers of two…
:  If z=1 Then                                                 :
:    d:=d-2^(i-1)                                              : and isolate remainder
:    Send "SET RGB eval(i-1) eval(r) eval(g) eval(b)"          : turn on corresponding LED
:  EndIf                                                       :
:EndFor                                                        : keep going, till all 8 bits done
:EndPrgm
```

Binary Trainer using RGB Array

<Green> subroutine:

```
Define gren(d)=                    : d holds the decimal number to be flashed in “green”
Prgm                               :
:Local i                           :All subroutine variables are kept local
:Send "SET RGB all 0 0 0"          :Reset all LEDs
:For i,1,8                         :Flash correct number 8 times
:     dispbin(d,"g")               :Call the binary display subroutine
:Wait 0.1                          : wait 100ms before
:Send "SET RGB all 0 0 0"          : turning off the display
:Wait 0.1                          : wait another 100ms before repeating the sequence
:EndFor                            :
:EndPrgm                           :
```

HMH 12/10/19

Binary Trainer using RGB Array

<red> subroutine:

```
Define red(a,r)=                    : a holds incorrect answer, r holds correct number
Prgm                                :
:Local i                            :all subroutine variables are kept local
:For i,1,8                          :8 flashes of
:  dispbin(r,"g")                   : the correct number in green
:Wait 0.1                           : and
:  dispbin(a,"r")                   : the incorrect number in red
:Wait 0.1                           :
:EndFor                             :
:EndPrgm                            :
```

HMH 12/10/19