1. Hammer: <http://www.codeskulptor.org/#user40_SSkvpa7lNirPsEp.py>

import simplegui

global str1

str1 = []

global str2

str2 = ''

def binary\_value(x):

a = []

b = []

while( x/2 != 1):

a.append(x%2)

x /= 2

if(x%2==0):

a.append(0)

else:

a.append(1)

a.append(1)

a.reverse()

#str1 = ''

if(len(a) < 7):

y = len(a)

z = 7 - y

for i in range(0, z):

b.append(0)

for j in range(0, y):

b.append(a[j])

else:

b = a

return b

def hamm(t):

x = 0

y = []

c = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

three = [0, 0, 0, 0]

five = [0, 0, 0, 0]

six = [0, 0, 0, 0]

seven = [0, 0, 0, 0]

nine = [0, 0, 0, 0]

ten = [0, 0, 0, 0]

eleven = [0, 0, 0, 0]

x = ord(t)

y = binary\_value(x)

c[2] = y[0]

c[4] = y[1]

c[5] = y[2]

c[6] = y[3]

c[8] = y[4]

c[9] = y[5]

c[10] = y[6]

if(c[2]==1):

three[0] = three[1] =1

if(c[4]==1):

five[0] = five[2] = 1

if(c[5]==1):

six[1] = six[2] = 1

if(c[6]==1):

seven[0] = seven[1] = seven[2] = 1

if(c[8]==1):

nine[0] = nine[3] = 1

if(c[9]==1):

ten[1] = ten[3] = 1

if(c[10]==1):

eleven[0] = eleven[1] = eleven[3] = 1

ze = three[0]+five[0]+six[0]+seven[0]+nine[0]+ten[0]+eleven[0]

on = three[1]+five[1]+six[1]+seven[1]+nine[1]+ten[1]+eleven[1]

tw = three[2]+five[2]+six[2]+seven[2]+nine[2]+ten[2]+eleven[2]

th = three[3]+five[3]+six[3]+seven[3]+nine[3]+ten[3]+eleven[3]

if(ze%2 == 1):

c[0] = ze%2

if(on%2 == 1):

c[1] = on%2

if(tw%2 == 1):

c[3] = tw%2

if(th%2 == 1):

c[7] = th%2

return c

def convert\_to\_string(s):

str1 = ''

for k in range(0, len(s)):

str1 = str1 + str(s[k])

return str1

def input(i):

m = convert\_to\_string(binary\_value(ord(i)))

str3 = " Code for " + str(i) + " before hamming: " + m + " and THE HAMMING CODE FOR " + str(i) + " is: " + convert\_to\_string(hamm(i))

return str3

def word\_input(w):

global str1

global str2

for i in range(len(w)):

str1.extend([input(w[i])])

str2 = "\n".join(str1)

print str2

if (len(w)>1):

str2 = " Check in Output window"

def draw(canvas):

canvas.draw\_text(str2, (50, 40), 15, 'Green')

f = simplegui.create\_frame("HAMMING",1000,125)

f.set\_draw\_handler(draw)

f.add\_input("Enter the letter ", word\_input, 50)

f.start()

1. De-hammer: <http://www.codeskulptor.org/#user40_vCCeBzHwPoAOkdg.py>

import simplegui

global s2

s2 = ''

global s3

s3 = "In order to get a text just delete the code in input block and enter the code for next letter in text "

def code\_input(c):

global s2

s1 = str(c)

print s1

c = 64\*(int(s1[2])) + 32\*(int(s1[4])) + 16\*(int(s1[5])) + 8\*(int(s1[6])) + 4\*(int(s1[8])) + 2\*(int(s1[9])) + (int(s1[10]))

s2 = s2 + chr(c)

print s2

print ''

def draw(canvas):

canvas.draw\_text(s2, (50, 40), 15, 'Green')

canvas.draw\_text(s3, (50, 75), 12, 'Blue')

f = simplegui.create\_frame("DE-HAMMING",1000,125)

f.set\_draw\_handler(draw)

f.add\_input("Enter the 11 digit binary Hamming code ", code\_input, 100)

f.start()

1. Error correction: <http://www.codeskulptor.org/#user40_b5WzQtQqmBJ6hdR.py>

import simplegui

global s2

s2 = ''

global s3

s3 = ''

def code\_input(c):

global s3

s1 = str(c)

counter = 0

one = [0, 0, 0, 0]

two = [0, 0, 0, 0]

three = [0, 0, 0, 0]

four = [0, 0, 0, 0]

five = [0, 0, 0, 0]

six = [0, 0, 0, 0]

seven = [0, 0, 0, 0]

eight = [0, 0, 0, 0]

nine = [0, 0, 0, 0]

ten = [0, 0, 0, 0]

eleven = [0, 0, 0, 0]

if(int(s1[0])==1):

one[0] = 1

if(int(s1[1])==1):

two[1] = 1

if(int(s1[2])==1):

three[0] = three[1] = 1

if(int(s1[3])==1):

four[2] = 1

if(int(s1[4])==1):

five[0] = five[2] = 1

if(int(s1[5])==1):

six[1] = six[2] = 1

if(int(s1[6])==1):

seven[0] = seven[1] = seven[2] = 1

if(int(s1[7])==1):

eight[3] = 1

if(int(s1[8])==1):

nine[0] = nine[3] = 1

if(int(s1[9])==1):

ten[1] = ten[3] = 1

if(int(s1[10])==1):

eleven[0] = eleven[1] = eleven[3] = 1

a1 = one[0] + two[0] + three[0] + four[0] + five[0] + six[0] + seven[0] + eight[0] + nine[0] + ten[0] + eleven[0]

a2 = one[1] + two[1] + three[1] + four[1] + five[1] + six[1] + seven[1] + eight[1] + nine[1] + ten[1] + eleven[1]

a4 = one[2] + two[2] + three[2] + four[2] + five[2] + six[2] + seven[2] + eight[2] + nine[2] + ten[2] + eleven[2]

a8 = one[3] + two[3] + three[3] + four[3] + five[3] + six[3] + seven[3] + eight[3] + nine[3] + ten[3] + eleven[3]

if(a1%2==1):

counter += 1

if(a2%2==1):

counter += 2

if(a4%2==1):

counter += 4

if(a8%2==1):

counter += 8

if(int(s1[counter-1])==1):

s2 = s1[0: counter-1] + '0' + s1[counter:]

elif(int(s1[counter-1])==1):

s2 = s1[0: counter-1] + '1' + s1[counter:]

elif(counter==0):

s2 = s1

if(s1==s2):

s3 = "Given code is correct code without errors"

elif(s1!=s2):

s3 = "Correct code is: "+s2

def draw(canvas):

canvas.draw\_text(s3, (50, 40), 15, 'Green')

f = simplegui.create\_frame("ERROR CORRECTION",500,125)

f.set\_draw\_handler(draw)

f.add\_input("Enter the 11 digit binary code ", code\_input, 100)

f.start()