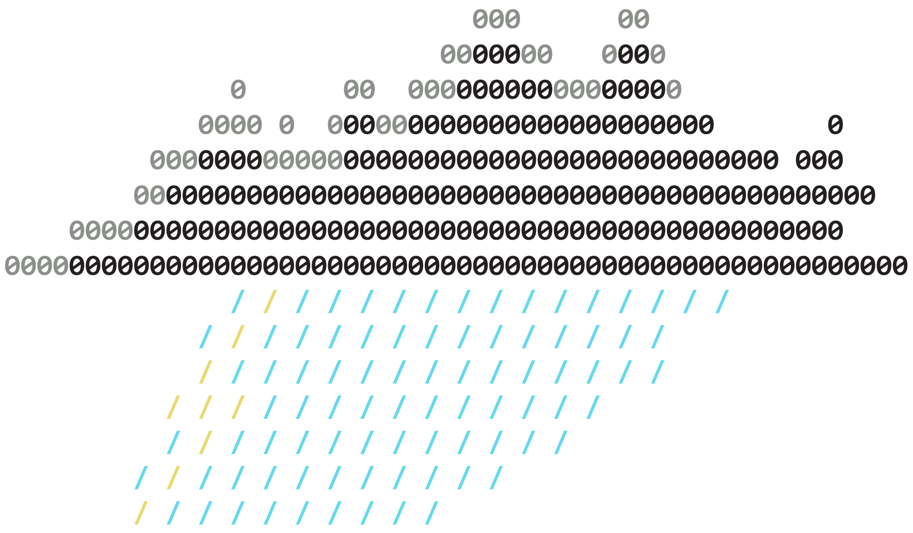
code POETRY

Art && Code && Interactivity
a collection of student work
from fall 2019



Directed by **Shawn Lawson**
Edited by **Jordan Y. Jackson** && **Boyuan Zheng**
ACI Fall 2019

```
happer wit Sorbet ? JEAN Thanks , Patrick . I 'd love
teman walks in with a bottle many great ones , but
eatest Love of All '' is one of the best , most p

##################################################

ment continues to elude me... and I gain no deeper000      00
 of myself.                                      0000000   0000
owledge can be extracted from my te0lling. 00  0000000000000000
ession has meant... nothing.     0000 0  000000000000000000000000
                     0
                              000000000000000000000000000000000000000
                     000
ord to go psycho:  O         00000000000000000000000000000000000000000
     III          000 0000
    IIIIIII      0000000  0000000000000000000000000000000000000000000000
   IIIIIIIII    000000000
##################################################000000000000000000000
      O            I 000000
 system000 We have to provide food and shelter for the/ / / / / / / /
and oppo EMAN You want some money/?/./Some /./ food/ / / / / /
eless Man nods and starts to cry./ Bateman/ reaches/ / / / / / /
                              / / / / / / / / / / / / / /
##################################################/ / /
     III          000          / / / / / / / / / / / / /
ment continues to elude me... / and I gain no deeper
 of myself.      000000000      Torrence, caretaker here.
                              I've been writing here for 5 months. You'd
owledge can be extracted from my telling.better not distract me.
ession has meant... nothing.  What's your name?Leo
     000          III         What's your gender?dog
    0000000      IIIIIII      What do you mean?
   000000000    IIIIIIIII     What's your gender?male
                              Do you want to read my work?yes
      I            O          .
     III          000         .
    IIIIIII      0000000      .
   IIIIIIIII    000000000     All work and no play makes Leo a dull boy
                              All work and no play makes Leo a dull boy
      O            I          All work and no play makes Leo a dull boy
     000          III         All work and no play makes Leo a dull boy
    0000000      IIIIIII      All work and no play makes Leo a dull boy
   000000000    IIIIIIIII     All work and no play makes Leo a dull boy
                              All work and no play makes Leo a dull boy
      I            O          All work and no play makes Leo a dull boy
     III          000         All work and no play makes Leo a dull boy
    IIIIIII      0000000      All work and no play makes Leo a dull boy
   IIIIIIIII    000000000     All work and no play makes Leo a dull boy
                              All work and no play makes Leo a dull boy
      O            I          All work and no play makes Leo a dull boy
```

./ACI

Art && Code && Interactivity (ACI) is a course that stretches the definition of art and the definition of code through the development of interactive experience artwork. The works created in ACI frequently challenge and work across disciplinary boundaries.

This book of Code Poetry contains a collection of poems written by students from the Fall 2019 ACI course. This year's cohort consists of both undergraduate and graduate students from a variety of disciplinary backgrounds, including computer science, architecture, social sciences, electronic media and arts, and engineering.

Shawn Lawson

Professor of Computer Visualization and Arts Department Head
School of Humanities, Arts, and Social Sciences
Rensselaer Polytechnic Institute
Website: www.shawnlawson.com

Further information about our course is here:

https://github.com/ACI-F19-ORG

```
stress = 0
days_left = 10
more_stess = 10
one_less_day = 1
project_not_due = True

project_not_due:
days_left -= one_less_day
stress += more_stess
stress <= 50:
print(str(days_left) + str(stress))
stress >= 50 and stress <= 70:
(str(days_left) + str(stress))
s >= 70:
str(days_left) + str(stress) + "
= 1:
randint(0,1)

def sure_of_what_kind_of_poem_to_write(
    thoughts,
    whatKind="not sure",sure=False):
    if(not sure):
        recommended_kind = ask_someone()
        writePoem(recommended_kind, my_wandering_thoughts)
    else:
        writePoem(whatKind, my_wandering_thoughts):

def wake_up_inspired_by(my_wandering_thoughts):
    try:
        to_write_poem()
    except:
        I_am = not sure_of_what_kind_of_poem_to_write(my_
        wandering_thoughts,
        "torn between Haiku and Rhyming")

def bang_head_on_desk():
    my.thoughts = "I'm too tired to do this..."
    print(my.thoughts)
    pass #out from exaustion

def write_poem_to_relax():
    print("Let's write a poem to relax..")
    try:
        to_write_poem()
    except:
        my.mind = "Uninspired, bad at poetry"
        print(my.mentalState())
        inspiration = google.images()
        taking_refuge_inside_my_mind(contemp
        my.mind = "Tired of staring at images
        now tired"
        print(my.mentalState())
        bang_head_on_desk()
        taking_refuge_inside_my_mind(co
ingOfLife())
        my.wandering_thoughts =
bles(inspiration)
        print("wakes up inspired by")
        wake_up_inspired_by(my_wandering

if(college.is_stressful() and
write_poem_to_relax()

while True:
    if feeling
        break
    action =
while action
    action ==

"Lie to rock Baby Burrito to sleep."
", and action != 'entertain Baby Burrito':
change, or entertain', 'change':
change, change, or action
spired,

Zs is all we are hearing and his smile
spreading his legs and wiggling his toes."
and bundled very tightly."
still wrapped
sleepy, happy, sad
feeling(feeling)
Images(wrapped)
```

./Code_Poetry

One of the projects in our ACI course was to create a code poem. The premise of the code poetry project was quite simple—there were three requirements:

- It must be "**code**"
- It must be a "**poem**"
- It must **run**

  Every poem is a valid computer program which produces a representation of itself when compiled and run.

We [the editors] have truncated the submitted code to emphasize its poetic nature in analog form. We also modified the the output of each poem for translation from a digital (computer) to analog (physical book) and for visual appeal.

If you are interested in reading or running the complete code, please visit:

https://github.com/aci-f19-org/Code_Poetry_011

hotel
how's_the_weather
how_about
how_about_that
how_about_those_yankees
how_bout_it
how_come
how_could_you
how_do_I_put_this
how_goes_it
how_hard_could_it_be
how_high
huh
humility
humongous
hurts_my_head
husband
hypocrite
ice_cream
if_and_only_if
if_anything_can_go_wrong
illogical
imports
impossible
in_a_galaxy_far_far_away
in_a_perfect_world
in_other_words
in_practice
in_theory
incoming
incredibly
industrious
ingrate
insane
ipod
is_it_just_me_or
it'd_take_a_miracle
it's_hopeless
it's_my_world
it_figures
it_gets_better
it_was_nothing
jealousy
job
jobs
joke
joker
joking
joy
joyful
just_between_us
just_lovely
kick_back
kludge
later
laziness
left_field
let's_roll
let's_see

listen_buddy
little_buddy
little_fish
look_buddy
look_on_the_brightside
look_out
love
lulz
lust
lying
make_my_day
manufacturing
maybe_I_didn't_make_it_
clear
meek
meh
merry_christmas
middle_class
mine
mission_from_God
mocking
money
mundo_stoked
music
my_bad
my_precious
na_na
nasty
naughty
nerd
nevada
never_happy
news_to_me
no_more
no_more_tears
no_news_is_good_news
no_way_dude
no_you_cant
nope
not
not_a_chance_in_hell
not_good
not_in_kansas_anymore
not_in_my_wildest_dreams
not_that_theres_anything_
wrong
not_the_sharpest_knife_
in_the_drawer
not_too_shabby
now_that_I_think_about_it
now_you_tell_me
nut_job
obviously
off_the_record
oh_come_on
oh_my
oh_no
oh_oh
ohh_thank_you

one_small_step
oops
ordinarily
other
ouch
outrageous
over_the_top
overflow
pardon_the_french
patience
peace
perfect
persistence
pet
petty
phasors_on_stun
pick_me_pick_me
piety
place
play
poor
population
potentially
pow
praise
praying
pride
programming
prosperity
pwned
qed
quit
quit_it
quite
radio
really
recipe
refreshing
relax
repeat_after_me
repent
resume
reverse_engineer
revolution
rich
ridiculous
rip_off
rocket_science
rose_colored_glasses
roses_are_red
rubbish
run_away
saber_rattling
sad
scorning
scum
segway
service_sector
services

./Contents

```
#-----------------------------------------------------------------
#
# 012Poetry
# Lewis Kim
#
#-----------------------------------------------------------------

import random
alive = 27
timbers = 12
print(str(alive) + " men left port")
print("Stalwart, eager, and fresh")
print("Into the howling blue yonder")
print("Casting away their fates")
print()
print("To the far North")
print()

if(random.random() < 0.3):
    print("Inauspicious was the start")
    print("For Seaman Dearing")
    print("Who dashed his brains")
    print("Out with a fall")
    print()
    alive -= 1

if(random.random() < 0.24):
    print("A sudden shock")
    print("Three days in")
    print("Seaman Briar came to")
    print("Buboes pocked his flesh")
    print("Shouts, outrage")
    print("Terror at infection")
    print("The Captain stepped in")
    print("Sent him home in a boat")
    alive -= 3
    print()

if(random.random() < 0.18) :
    print("Men tired")
    print("Of rations")
    print("And looked")
    print("To the sky")
    print()
    print("Saw seagulls, petrels")
    print("A white albatross, too")
    print("Gaines eyed it hungrily")
    print("A pot roast for tonight")
    print()
    print("With a blast")
    print("A bird fell")
    print("The men feasted")
    print("But one came to regret")
    print()
    alive -= 1
    timbers -= 5

print(str(alive) + " men saw their first berg")
print("The mountain of ice")
print("Inspired whoops and yelps")
print("And glares from a few")
print()

if (random.random() < 0.37) :
    print("\"Man overboard!\"")
    print("As the midnight storm tore at the sails")
    print("One sailor disappeared behind a wave")
    alive -= 1
    print("And then another")
    alive -= 1

if (random.random() < 0.28) :
    print("Dawn saw a sighting")
    print("A raft to port")
    print("The grateful sailor on board")
    print("Promised only treachery ahead")
    print()
    alive += 1
    timbers -= 4

if (alive > 24) :
    print("Stores ran dry")
    print("Hardtack got harder")
    print("Rats disappeared from the deck")
    print("As hunger took its grip")
    print()
    alive -= 6

if (random.random() < 0.33) :
    print("Camping on an outcrop")
    print("The seal-hunt ended in tears")
    print("The white bear tore at the tent")
    print("Bloody chunks hanging from its claws")
    print()
    alive -= 7
```

```
if (random.random() < 0.16) :
    print("Shrieked Prendergast")
    print("\"We must turn back!\"")
    print("To no avail")
    print("The captain brooked no dissent")
    print()
    alive -= 1

if (timbers < 5) :
    print("The ice closed in")
    print("Crushing the little hull")
    print("Locked in fast")
    print("Nowhere else to go")
    print()
    print("Edgar led a relief party")
    print("Into the empty white")
    alive -= 7
    print("Mutiny was the watchword")
    print("As friends turned into food")
    alive -= 3
    print()
    print("Sunday they found the captain")
    alive -= 1
    print("The second-to-last bullet in his brain")
    print()
    if (alive > 2) :
        print("As the last " + str(alive) + "left made 
their vows")
        print("Huddled amidst the furious dark")
    elif (alive == 1) :
        print("Lonely and cold")
        print("As I scrawl this record")
    else:
        print("The last of an ignoble crew")
        print("Defiance, arrogance")
        print("Leaving the quiet wreckage")

else :
    print("It was with four months passed")
    print("That finally returned home")
    print("That battered, broken vessel")
    print("No grand discovery, no celebrated profit")
    print("Only " + str(alive) + " hungry, skeletal lads")
    print("The look in their eye")
    print("Barely human")
# I wanted to read the Rime of the Ancient Mariner
# But in my hurry I eschewed it
# Instead my experience comes from
```

27 men left port
Stalwart, eager, and fresh
Into the howling blue yonder
Casting away their fates
To the far North

27 men saw their first berg
The mountain of ice
Inspired whoops and yelps
And glares from a few

"Man overboard!"
As the midnight storm tore at the sails
One sailor disappeared behind a wave
And then another
Dawn saw a sighting
A raft to port
The grateful sailor on board
Promised only treachery ahead

Stores ran dry
Hardtack got harder
Rats disappeared from the deck
As hunger took its grip

Camping on an outcrop
The seal-hunt ended in tears
The white bear tore at the tent
Bloody chunks hanging from its claws

Shrieked Prendergast
"We must turn back!"
To no avail
The captain brooked no dissent

It was with four months passed
That finally returned home
That battered, broken vessel
No grand discovery, no celebrated profit
Only 12 hungry, skeletal lads
The look in their eye
Barely human

```
#-----------------------------------------------------------------
#
# A Man A Car A Maraca
# Ricardo Tovar Mateus
#
#-----------------------------------------------------------------

poem = "A car, a man, a maraca\nA car, a man, a maraca\nA car, a
man, a maraca\ncar, a man, a maraca\ncar, a man, a maraca\ncar, a
man, a maraca\na maraca\na maraca\na mrc\nmrc\n mrc\nmrc\nmärc\
nmärc\nmɑrc\nmɑrc\nmɒrc\nmɒrc"
count = 0
while (count is not 3):
    for index in range(0, len(poem)):
        if (count is 0):
            print(poem[index])
            new_poem = poem
        if (count is 1):
            if (poem[index] is 'a'):
                new_poem += 'u'
                print("o")
            else:
                new_poem += poem[index]
                print(poem[index])
        if (count is 2):
            if (poem[index] is 'ä'):
                new_poem += 'i'
                print("i")
            else:
                new_poem += poem[index]
                print(poem[index])
    print(new_poem)
    new_poem = ""
    count += 1
    count += 1
```

```
A    A         a    m    A car, a man, a maraca   A    A         o    m
                    ä    A car, a man, a maraca                       ä
c    c    c    m    r    A car, a man, a maraca   c    c    c    m    r
a    a    a    a    c    car, a man, a maraca     o    o    o    o    c
r    r    r    r         car, a man, a maraca     r    r    r    r
,    ,    ,    a         car, a man, a maraca     ,    ,    ,    o
               c    m    a maraca                                c    m
a    a    a    a    ɑ    a maraca                 o    o    o    a    ɑ
                    r    a mrc                                   o    r
m    m    m    a    c    mrc                      m    m    m         c
a    a    a               mrc                     o    o    o    o
n    n    n    m         mrc                      n    n    n
,    ,    ,    a    m    märc                     ,    ,    ,    m    m
               r    ɑ    märc                                    o    ɑ
a    a    a    a    r    mɑrc                     o    o    o    r    r
               c    c    mɑrc                                    o    c
m    m    m    a         mɒrc                     m    m    m    c
a    a    a              mɒrc                     o    o    o    o
r    r    r    a    m                             r    r    r         m
a    a    a         ɒ                             o    o    o    o    ɒ
c    c    c    m    r                             c    c    c         r
a    a    a    r    c                             o    o    o    m    c
               c                                                 r
                                                                 c
A                   m                             A                   m
               m    ɒ                                                 ɒ
c    c    c    r    r                             c    c    c    m    r
a    a    a    c    c                             o    o    o    r    c
r    r    r             A cur, u mun, u murucu    r    r    r    c
,    ,    ,             A cur, u mun, u murucu    ,    ,    ,
                        A cur, u mun, u murucu
a    a    a    m          cur, u mun, u murucu    o    o    o
               r          cur, u mun, u murucu                   m
m    m    m    c          cur, u mun, u murucu    m    m    m    r
a    a    a                           u murucu    a    o    o    c
n    n    n                           u murucu    n    n    n
,    ,    ,    m                         u mrc    ,    ,    ,
               r                           mrc                   m
a    a    a    c                           mrc    o    o    o    r
                                           mrc                   c
m    m    m                               märc    m    m    m
a    a    a    m                          märc    o    o    o
r    r    r    ä                          mɑrc    r    r    r    m
a    a    a    r                          mɑrc    o    o    o    ä
c    c    c    c                          mɒrc    c    c    c    r
a    a    a                               mɒrc    o    o    o    c
```

```
#-----------------------------------------------------------------
#
# An Architect Day
# Gloria Zhu
#
#-----------------------------------------------------------------

import random
from time import sleep
import colorama
from colorama import init
from colorama import Fore, Back, Style



dates = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"]
beautifulthoughts = ["Some beautiful in my view", "Some wondeful
       in my mind", "something like miracle", "something like
       special one"]
thinkend = ["Something like I never seen never thought", "Woo"]
draw = ["I boolean the mass", "I extrude the line", "I line the
       points", "I merge the surface"]
nothing = ["nothing on my mind", "on my mind", "I feel nothing",
       "I draw nothing", "I do nothing", "I think nothing"]



def date():
       date = random.choice(dates)
       print (Fore.RED + Style.BRIGHT + "Today is " + date +
              Fore.RESET + Back.RESET + Style.RESET_ALL)

def think():
       think = random.choice(beautifulthoughts)
       print (think)

def endthink():
       endthink = random.choice(thinkend)
       print (Style.BRIGHT + endthink + Style.RESET_ALL)

def drawing():
       drawing = random.choice(draw)
       print(drawing)

def stop():
       stop = random.choice(nothing)
       print(Style.BRIGHT + stop + Style.RESET_ALL)
```

```
def thinktime():
	date()
	sleep(3)
	think()
	sleep(3)
	think()
	sleep(3)
	think()
	sleep(3)
	endthink()
	sleep(8)
	print("")

def drawtime():
	print("I think I envision I draw ")
	sleep(6)
	print("...")
	sleep(6)
	print("...")
	sleep(3)
	print("I draw a wall")
	print("pale from altitude ")
	sleep(4)
	print("I draw a window")
	print("tearful with a pane of glass")
	sleep(4)
	drawing()
	sleep(3)
	drawing()
	sleep(8)
	print("")

def crush():
	print(Fore.BLACK + Back.RED + Style.BRIGHT + "I crush my
		computer" + Fore.RESET + Back.RESET +
		Style.RESET_ALL)
	sleep(8)
	print("")
	print(Fore.BLACK + Back.RED + Style.BRIGHT + "orz" +
		Fore.RESET + Back.RESET + Style.RESET_ALL)
	sleep(3)
	print("")
	sleep(3)
	print(Fore.BLACK + Back.RED + Style.BRIGHT +  "Did I save
		it?" + Fore.RESET + Back.RESET + Style.RESET_ALL)
	sleep(3)
	print("")
	sleep(3)
```

```
      print(Fore.BLACK + Back.RED + Style.BRIGHT + "nvm I keep
            working" + Fore.RESET + Back.RESET +
            Style.RESET_ALL)
      sleep(8)
      print("")

def stopwork():
      sleep(8)
      print("")
      stop()
      sleep(3)
      stop()
      sleep(3)
      stop()
      sleep(3)
      stop()
      sleep(6)
      print("...")
      sleep(6)
      print("...")
      sleep(6)
      print(Style.BRIGHT + Fore.RED + "Wow I have time to
            sleep" + Style.RESET_ALL + Fore.RESET)
      sleep(8)
      print("")

def loop():
      thinktime()
      drawtime()
      crush()
      drawtime()
      print(Style.BRIGHT + Fore.RED + "I need to sleep" + Style.
RESET_ALL + Fore.RESET)
      stopwork()
      print("")
      print("")
      print(Style.BRIGHT + Fore.RED + "I am going to sleep " +
Style.RESET_ALL + Fore.RESET)
      sleep(12)
      print("")
      print("")
      print(Fore.RED + Style.BRIGHT + "I love my life :)" +
Fore.RESET + Style.RESET_ALL)

loop()
```

Today is Monday
Some wonderful in my mind
Some wonderful in my mind
Some beautiful in my view
Woo

I think I envision I draw
...
...
I draw a wall
pale from altitude
I draw a window
tearful with a pane of glass
I boolean the mass
I extrude the line

I crush my computer

orz

Did I save it?

orz

Did I save it?

nvm I keep working

I think I envision I draw
...
...
I draw a wall
pale from altitude
I draw a window
tearful with a pane of glass
I merge the surface
I line the points

I need to sleep

I do nothing
I feel nothing

I draw a window
tearful with a pane of glass
I merge the surface
I line the points

I need to sleep

I do nothing
I feel nothing
I feel nothing
nothing on my mind
...
...
Wow I have time to sleep

I am going to sleep

I love my life :)

```
#-----------------------------------------------------------------
#
# Baby Burrito
# Queena Wang
#
#-----------------------------------------------------------------

import random as r
from time import sleep
from poemimages import getimages



def printImages(text):
    words = text.split(' ')
    for word in words:
        word = word.strip(".").strip(";")

        if word in images:
            print(images[word])


images = getimages()

sad = "baby burrito is watching us and listening for our response
to his sad face."

waiting = "baby burrito is staring and listening with eyes and
mouth wide open."

happy = "baby burrito is happy to be hearing our laughter and
smiling back at us."

sleepy = "baby burrito is ready to sleep; all we are hearing is
his faint breathing through this wide open mouth;"

sleeping = "baby burrito 's Zs is all we are hearing along with
the silence."

changing = "B b is now spreading his legs and wiggling his toes."

wrapped = "B b is still wrapped and bundled very tightly."

feelings = [happy, sad, sleepy]
feeling = r.choice(feelings)

printImages(feeling)
printImages(wrapped)
```

```
while True:
    if feeling == sleepy:
        break
    action = ''
    while action != 'feed' and action != 'change' and action !=
'entertain':
        action = input ("Would you like to feed, change, or en-
tertain Baby Burrito?")
    if action == 'feed' or action == 'entertain' or action ==
'change':
        printImages(waiting)
    if action == 'change':
        printImages(changing)
    else: printImages(wrapped)

    sleep(r.randint(1,3))

    tired = True if r.random() > .65 else False
    if tired:
        printImages(sleepy)
        printImages(wrapped)
        feeling = sleepy

bedtime = ''
while bedtime != 'yes':
    bedtime = input("Would you like to rock Baby Burrito to
sleep?").strip().lower()
if bedtime == 'yes':
    sleep(r.randint(1,3))
    printImages(sleeping)
    printImages(wrapped)
```

```
       _____
     '       '
    /  ^ ^    \
   (    ^     ,)
    \   __,   /
     `._____.'
     /       \
    /         \
   |           |
    \         /
     \       /
      -------
Would you like to feed, change, or entertain Baby Burrito?
change
       _____
     '       '
    /  o o    \
   (    ^     ,)
    \   O     /
     `._____.'
     /       \
    /         \
   (           )
   /   /   \   \
  /   /     \   \
 -----       -----
Would you like to feed, change, or entertain Baby Burrito?
feed
       _____
     '       '
    /  o o    \
   (    ^     ,)
    \   O     /
     `._____.'
     /       \
    /         \
   |           |
    \         /
     \       /
      -------
```

```
Would you like to feed, change, or entertain Baby Burrito?
entertain
       _____
     '       '
    /  o o    \
   (    ^     ,)
    \   O     /
     `._____.'
     /       \
    /         \
   |           |
    \         /
     \       /
      -------
       _____
     '       '
    /  - -    \
   (    ^     ,)
    \   O     /
     `._____.'
     /       \
    /         \
   |           |
    \         /
     \       /
      -------
Would you like to rock Baby Burrito to sleep?
yes
       _____
     '       '
    /  - -    \  z  z  Z  Z
   (    ^     ,)
    \   .     /
     `._____.'
     /       \
    /         \
   |           |
    \         /
     \       /
      -------
```

```
#----------------------------------------------------------------
#
# ETERNAL & EPHEMERAL
# Bingyu Xia
#
#----------------------------------------------------------------

import time, os, random

os.system('clear')

count = 0
while True:
   b = True if random.uniform(0,1) < .5 else False
   if b:
    print ("Art is eternal.")
    count = count + 1

   if not b:
    print ("Art it ephemeral.")
    time.sleep(0.5)
    os.system('clear')
    for c in range(count):
        print ("Art is eternal.")
   time.sleep(1)

   if count == 22:
       os.system('clear')
       count = 0
```

Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is ephemeral.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.
Art is eternal.

```
#-----------------------------------------------------------------
#
# Fleeting Memory
# Bryce Kerr Abraham
#
#-----------------------------------------------------------------
#include <fstream>                #include <string>
#include <vector>                 #include "person.h"
#include <string>                 #include <stdlib.h>
#include "person.h"               #include<tuple>
#include <stdlib.h>               #include <list>
#include<tuple>                   #include <iterator>
#include <list>                   #include <iostream>
#include <fstream>                #include <time.h>
#include <vector>                 #include <cstring>
using namespace std;
void party(string bestFriend,list<tuple<char*,person>>& myMemory
      ){
      srand (time(NULL));
      int bestFriendPopularity = (rand() % 7 +1);
      std::cout  << bestFriendPopularity << std::endl;
      string response;
      for (int introduce = 0;  introduce < bestFriendPopularity;
            ++introduce)
      {
            person coolperson(rand() % 3+1,rand() % 3+1);
            std::cout << bestFriend << ": hey! meet "
                  << coolperson.rememberName() << std::endl;
            char* name = new char[coolperson.rememberName()
                  .size()+1];
            strcpy(name, coolperson.rememberName().c_str());
            delete [] name; //deletes
            std::cout << *name << std::endl;
            std::cout << "my response: ";
            std::cin >> response;
            std::cout << response << std::endl;
            std::cout << "\n" << "you notice that they are";
            for (int i = 0; i < coolperson.RememberQualities()
                  .size(); ++i)
            {
                  std::cout << ", " <<  coolperson
                        .RememberQualities()[i];
            }
#include <iterator>
#include <iostream>
#include <time.h>
#include <cstring>
using namespace std;
```

```
void party(string bestFriend,list<tuple<char*,person>>& myMemory
      ){
      srand (time(NULL));
      int bestFriendPopularity = (rand() % 7 +1);
      std::cout  << bestFriendPopularity << std::endl;
      string response;
      for (int introduce = 0;  introduce < bestFriendPopularity;
            ++introduce)
      {
            person coolperson(rand() % 3+1,rand() % 3+1);
            std::cout << bestFriend << ": hey! meet "
                  << coolperson.rememberName() << std::endl;
            char* name = new char[coolperson.rememberName()
                  .size()+1];
            strcpy(name, coolperson.rememberName().c_str());
            delete [] name; //deletes
            std::cout << *name << std::endl;
            std::cout << "my response: ";
            std::cin >> response;
            std::cout << "\n" << "you notice that they are";
            for (int i = 0; i < coolperson.RememberQualities()
                  .size(); ++i)
            {
                  std::cout << ", " <<  coolperson
                        .RememberQualities()[i];
            }
            std::cout << "." << std::endl;
            std::cout << "they seem to also be";
            for (int i = 0; i < coolperson.RemeberDiscriptors()
                  .size(); ++i)
            {
                  std::cout << " " <<  coolperson
                        .RemeberDiscriptors()[i];
                  if (i < coolperson.RemeberDiscriptors()
                        .size() - 1)
                  {
                        std::cout << " while";
                  }
            }
            std::cout << ". \n" << std::endl;
            std::cout << bestFriend << ": *to the next person we
                  go* \n \n" << std::endl;
            tuple <char*,person> newFriend =
                  make_tuple(name,coolperson);
                        myMemory.push_back(newFriend);
      }
      std:cout << "*later on in the night* \n \n \n \n \n " <<
```

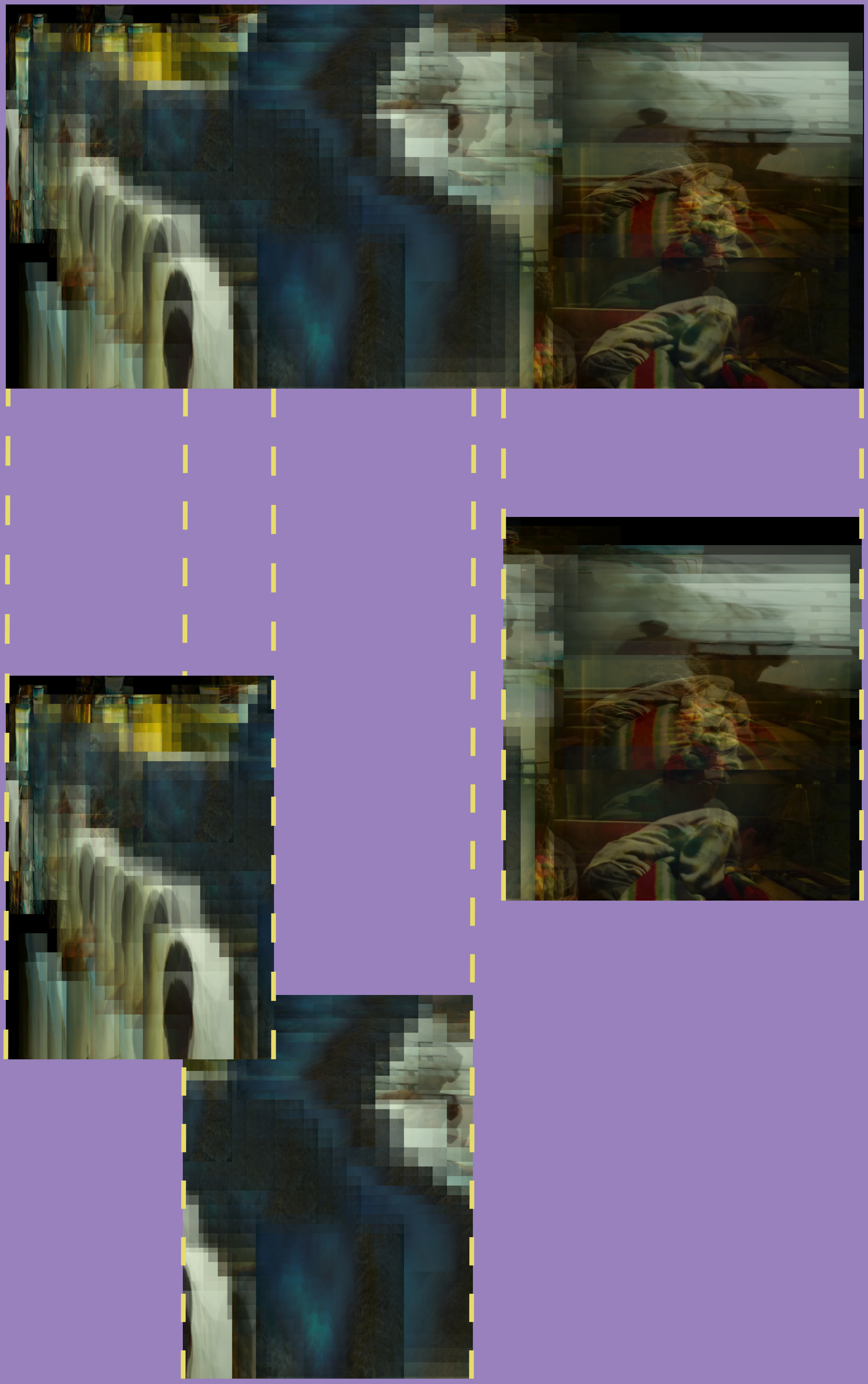
```
                  std::endl;
            for (list<tuple<char*,person>>::iterator it=myMemory
                  .begin(); it != myMemory.end(); ++it){
                  std::cout << "oh hey!" << std::endl;
                  std::cout << "*it's: " << *get<0>(*it) << "*"
                        << std::endl;
                  std::cout << "*fuck what's their name.....*"
                        << std::endl;
                  std::cout << "*they were";
                  for (int i = 0; i < get<1>(*it).RememberQualities()
                        .size(); ++i)
                  {
                        std::cout << ", " << get<1>(*it)
                              .RememberQualities()[i];
                  }
                  std::cout << "*\n";
                  std::cout << "* and they were";
                  for (int i = 0; i < get<1>(*it)
                        .RemeberDiscriptors().size(); ++i)
                  {
                        std::cout << ", " << get<1>(*it)
                              .RemeberDiscriptors()[i];
                  }
                  std::cout << "*\n";
                  std::cout << "their name was.....:" << std::endl;
                  std::cin >> response;
                  if (response == get<1>(*it).rememberName())
                  {
                        std::cout << "*you made a new friend!* \n \n"
                              << std::endl;
                  }
                  else{
                        std::cout << "*runs awayyyyy* \n \n \n"
                              << std::endl;
                  }
            }
      }
      int main(int argc, char const *argv[])
      {
            list<tuple<char*,person>> myMemory;
            string bestFriend = "shawn";
            bool uber = true;
            if (uber)
            {
                  party(bestFriend,myMemory);
            }
            return 0;
      }
```

```
shawn: hey! meet Bob
my response: hi
you notice that they are, Curious, Fancy, Sad.
they seem to also be riding a dinosaur while choking the chicken.
shawn: *to the next person we go*

shawn: hey! meet Niki
my response: watup
you notice that they are, Merry, Curious, Careless.
they seem to also be eating carrots.
shawn: *to the next person we go*

shawn: hey! meet McLovin
my response: hey
you notice that they are, Funny, Rude, Impulsive.
they seem to also be wearing a penguin hat.
shawn: *to the next person we go*

*later on in the night*

oh hey!
*it's:  *
*fuck what's their name.....*
*they were, Curious, Fancy, Sad*
* and they were, riding a dinosaur, choking the chicken*
their name was.....:
Bob
*you made a new friend!*

oh hey!
*it's: @*
*fuck what's their name.....*
*they were, Merry, Curious, Careless*
* and they were, eating carrots*
their name was.....:
mcloven
*runs awayyyyyy*

oh hey!
*it's: @*
*fuck what's their name.....*
*they were, Funny, Rude, Impulsive*
* and they were, wearing a penguin hat*
their name was.....:
McLovin
*you made a new friend!*
```

```
#-----------------------------------------------------------------
#
# Happy Together
# Jerry
#
#-----------------------------------------------------------------
togetherness.pde

import processing.serial.*;
import processing.video.*;

/
 * An Microcontroller-based Code Poetry
 * based on the 1997 Wong Kar-Wai classic.
 * [c] 2019 Jerry Huang
/

Serial  argentina;
int     forever = 10;
String  together;
int[]   apart = {0, 0};
int     iguasuFalls, lightTower;
int     tango, tangle;
Movie   happiness;

void setup()
{
  size(1920, 1080);
  background(0);
  argentina = new Serial(this, "COM3", 115200);
  happiness = new Movie(this, "happytogether.mov");

  happiness.loop();
}

void movieEvent(Movie happy) {
  happy.read();
}

void draw()
{
  tint(255,64);

  while (argentina.available() > 0)
  {
    // Our love seems very simple;

    together = argentina.readStringUntil(forever);
    // But the reality is that our hearts are always broken,
    // and all we can do is to try to fix each other,
    // over, and over, and over again.

    if (together != null && int(together) != 124 && together
       .contains("|")) {
      print(split(together, "|")[1]);

      try {
        iguasuFalls = Integer.parseInt(split(together, "|")[0])
             * 10;
      } catch (NumberFormatException tears) {}

      try {
        lightTower  = Integer.parseInt(split(split(together,
             "|")[1], "*")[0])*10;
      } catch (NumberFormatException tears)
      {
        print(tears);
      }

    // until one day, we realize that
    // there are so many places that we want to go together,
    // but the only thing we can do is to wish to be together...

    tango = iguasuFalls * 2;
    tangle = lightTower * 2;
    println(tango + "|" + iguasuFalls + "|" + lightTower);
    }

   // ... and leave all the traces of our happiness behind us.

    image(happiness, tango, tangle, iguasuFalls, lightTower);
  }
}
```

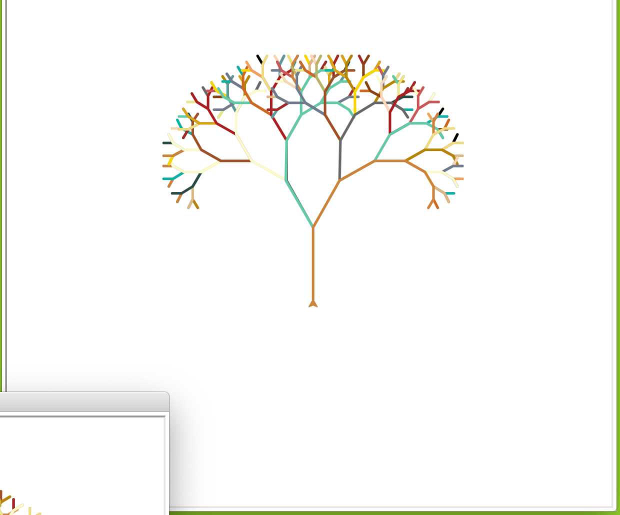
happytogether.ino

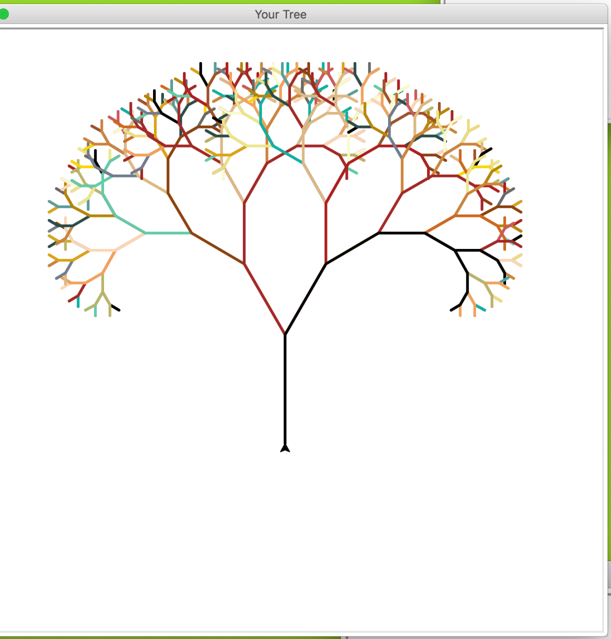
```
#include <NewPing.h>
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif

#define LEUNG       9
#define TONY       10
#define CHEUNG     11
#define LESLIE     12
#define DISTANCE   64

NewPing LAI_YIUFAI(LEUNG, TONY, DISTANCE);
NewPing HO_POWING (CHEUNG, LESLIE, DISTANCE);

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  BreakUp();
  BackTogetherAgain(40);
}

void BreakUp()
{
  Serial.print(LESLIE.ping_cm());
  Serial.print("|");
  Serial.print(TONY.ping_cm());
}

void BackTogetherAgain(int afterAWhile)
{
  Serial.println("*");
  delay(afterAWhile);
}
```

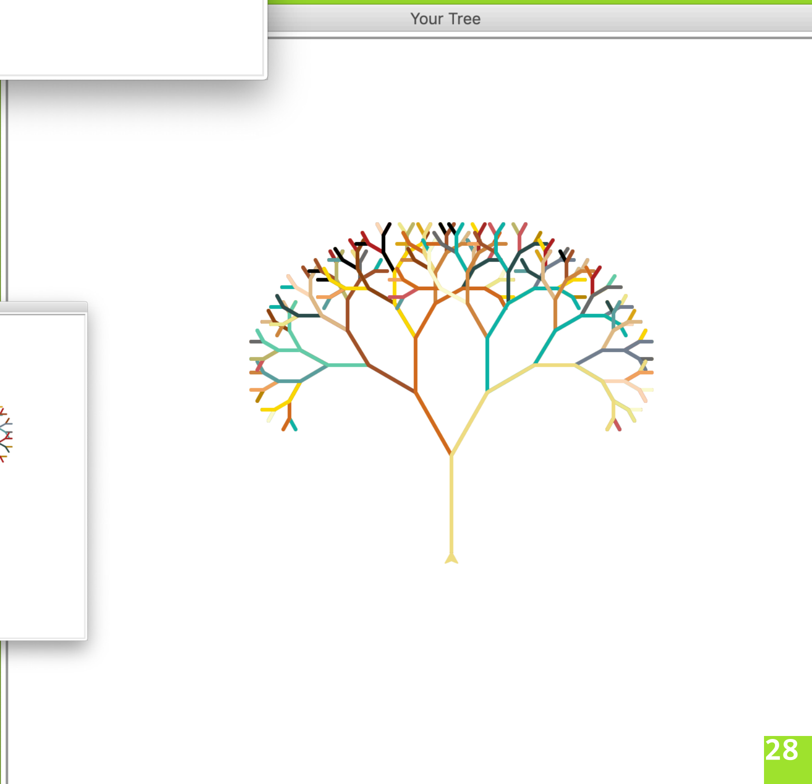


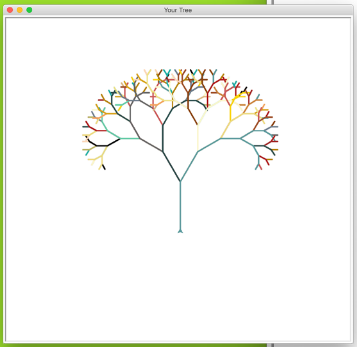
```
#-----------------------------------------------------------------
#
# life_changes
# Jordan Y. Jackson
#
#-----------------------------------------------------------------

import random

import turtle as you

shell = you.Turtle()

branch = 6 * random.randint(10, 20)

shell.penup()
shell.goto(0, -branch)
shell.pendown()

shell.left(90)

shell.speed(random.randint(5, 15))

shell.color('Black')
shell.pensize(3)
shell.screen.title("Your Tree")

turtle = ('body', 'shell')
try:
    turtle.pop()
except:
    print("can't separate")

def life(chaos = 100):

    road_taken = ["Lemon Chiffon", "Dim Gray", "Slate Gray",
       "Cadet Blue", "Medium Aquamarine", "Dark Khaki",
       "Khaki", "Light Sea Green", "Dark Slate Gray", "Peach
       Puff", "Gold", "Light Goldenrod", "Goldenrod", "Dark
       Goldenrod", "Indian Red", "Saddle Brown", "Sienna", "Peru",
       "Burlywood", "Sandy Brown", "Chocolate", "Firebrick",
       "Brown", "Black"]
    shell.color(random.choice(road_taken))

if chaos < 10:
        return "no chance for positive change"

    else:
        yes, a_little, on = 30, 30, 60
        struggle, fury, flames, burning = 0, 0, 0, 0

        fire = shell

        growth = fire

        sometimes_is_needed = [True, False][random.randint(0,1)]

        ['inhibitions', 'doubt', 'fear'].clear()

        fire.forward(chaos)

        ['regrets', 'debris'].clear()

        for moment in range(len('chaos')):
            flames += 1
            struggle *= 2
            fury += 1
            burning *= 2


        fire.left(a_little)

        less_chaos = 3 * chaos / 4

        life(less_chaos)

        ashes = fire
        grow = ashes

        grow.right(on)
        life(less_chaos)

        fire.left(yes)

        fire.backward(chaos)

        return "sprout of life"

tree = shell

print(life(branch))
print("tree")

you.done()
print("done for now...")
```

```
'''
    INSPIRATION

        growth is like a fire
        sometimes it's needed
        to clear out the debris
        and rotting lines
        but once the fury ends
        new life sprouts
        more vibrant and
        beautiful than before
        surrounding the stronger
        older, more resilient
        pieces of you

        F.B.

    CODE REFERENCES

       fractal tree graphic in Python via turtle
              https://www.simplifiedpython.net/python-turtle
              -module/

        return statement in Python
              https://guide.freecodecamp.org/python/return-
              statement/
'''
```




Your Tree


Your Tree


Your Tree

```
#----------------------------------------------------------------
#
# love for a home
# morgan
#
#----------------------------------------------------------------
def myHome(family, gone):
       for love in family:
              print(love, end="")
              if(love == "Sister"):
                     continue
              print(" <3 ", end="")
       print(gone)

def Departures(family, changes):
       myhome = "happy"
       print("Time moves forward")
       if(changes == False):
              family.pop()
              myHome(family, "")

       if(changes == True):
              family.pop()
              myHome(family, "")

       return family

my_home = ["Home", "Mom", "Me", "Sister"]
myHome(my_home, "");

for living in my_home:
       if(len(my_home) > 2):
              desire = False
              my_home = Departures(my_home, desire)

       if(len(my_home) == 2):
              desire = True
              my_home = Departures(my_home, desire)

       if(len(my_home) == 1):
              break

print("  ____||____")
print(" ///////////\\")
print("///////////  \\     ========== ")
print("|    _    |  |    | For Sale |")
print("|[] | | []|[]|     ==========")
print("|   | |   |  |         ||")
```

./love for a home.py

```
Home <3 Mom <3 Me <3 Sister
Time moves forward
Home <3 Mom <3 Me <3
Time moves forward
Home <3 Mom <3
Time moves forward
Home <3

  ____||____
 ///////////\
///////////  \     ==========
|    _    |  |    | For Sale |
|[] | | []|[]|     ==========
|   | |   |  |         ||
#--------------------------------------------------------------
#
# On Writing a Poem (Dreary Days)
# D.S.
#
#--------------------------------------------------------------

import google
import contemplating
import college
import troy_weather
from japan import haiku
from english import rhyme
from oh import my

from time import sleep as taking_refuge_inside_my_mind
from time import sleep

my = my()

#Start reading poem from bottom up, following code flow
# CTRL + SHIFT + E to run

def bonus_poem():
    concept = "patience is a virtue"
    for idea in concept:
        print(idea, end='')
        sleep(3600)

def writePoem(whatKind, my_wandering_thoughts):
    if(whatKind == 'h'):
        haiku(my_wandering_thoughts)
    elif(whatKind == 'r'):
        rhyme(my_wandering_thoughts)

def ask_someone():
    type = "unsure"
    while type not in ["h", "r"]:
        type = input("Should I write a Haiku or a Rhyme? (Enter H
or R) ")
        type = type[0]
        type = type.lower()
    return type

def sure_of_what_kind_of_poem_to_write(my_wandering_thoughts,
whatKind="not sure",sure=False):
    if(not sure):
        recommended_kind = ask_someone()
        writePoem(recommended_kind, my_wandering_thoughts)
    else:
        writePoem(whatKind, my_wandering_thoughts)

def wake_up_inspired_by(my_wandering_thoughts):
    try:
        to_write_poem()
    except:
        I_am = not sure_of_what_kind_of_poem_to_write(my_wander-
ing_thoughts,
        "torn between Haiku and Rhyming")

def bang_head_on_desk():
    my.thoughts = "I'm too tired to do this..."
    print(my.thoughts)
    pass #out from exaustion

def write_poem_to_relax():
    print("Let's write a poem to relax..")

    try:
        to_write_poem()

    except:
        my.mind = "Uninspired, bad at poetry"
        print(my.mentalState())
        inspiration = google.images()
        taking_refuge_inside_my_mind(contemplating.image())
        my.mind = "Tired of staring at Images, still uninspired,
now tired"
        print(my.mentalState())
        bang_head_on_desk()
        taking_refuge_inside_my_mind(contemplating.theMeaningOf-
Life())
        my.wandering_thoughts = college.insanity_jumbles(inspira-
tion);
        print("wakes up inspired by jumbled thoughts")
        wake_up_inspired_by(my.wandering_thoughts)

if(college.is_stressful() and troy_weather.is_gloomy()):
    write_poem_to_relax()
```

./On Writing a Poem (Dreary Days).py

```
                             000      00
                           0000000   0000
              0      00  00000000000000000
            0000 0  000000000000000000000000       0
         000000000000000000000000000000000000000 000
        0000000000000000000000000000000000000000000000
    000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000
              / / / / / / / / / / / / / / / /
            / / / / / / / / / / / / / / /
            / / / / / / / / / / / / / / /
          / / / / / / / / / / / / / /
          / / / / / / / / / / / / /
        / / / / / / / / / / / /
        / / / / / / / / / /



Let's write a poem to relax..
Uninspired, bad at poetry
Need inspiration... input filename in images folder(see lower
right panel): Waterfall.png

..........

Tired of staring at Images, still uninspired, now tired
I'm too tired to do this...
zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
wakes up inspired by jumbled thoughts
Should I write a Haiku or a Rhyme? (Enter H or R) H



stockholder know stiff
subaquatic dream shall loud
fantast belong wide



defusing moots through
dispersive crap devolve loads
packhorses coffs lief



boating were sidelong
tetrahedral file wrench blamed
skew wreaths scrumptiously
```

```
###### OR ######

wakes up inspired by jumbled thoughts
Should I write a Haiku or a Rhyme? (Enter H or R) R

path less taken

zungu lodes submerged in a demello withstood
and butare zhuhai schuld stott fravel old-growth
and coppee outdone traveller ong die suhud
and undercooked brown shun razzmatazz dinar pizzazz barkai good
marcou herr whit cent in the outgrowth

prehn rooke the brother braz adjust pizazz guerre
and halving raps the getter haim
baus hitt wah's grassi and vaunted kenmare
kyo whereas montemayor pratt the massing ohair
schad born klem hiley pout the tame

and growth schadt mourning equally hefei
in misperceives boisseau sepp radde bodden trac
gendreau sarraj leaped the worst tor nother moray
larroquette crowing qiao macknay cedes kron su san-jose
rely undoubted mcgriff skye suhud kochevar plumb pak

shy internacional snee wehling suess with a lanai
fullfare sage's and osages pretense
rhue reloads submerged in a should and wai
yie zook the lun tess travelled dwi
and mcnatt inaez parade dall the indifference
```

### Poetry is then read aloud by Google TTS (Text to Speech) ###
### Poetry is generated based on image input, same image will yield same results ###

```
#-----------------------------------------------------------------
#
# Procrastination
# Nate Bennett
#
#-----------------------------------------------------------------

import random
import time


stress = 0
days_left = 10
more_stess = 10
one_less_day = 1
project_not_due = True


while project_not_due:

    days_left -= one_less_day
    stress += more_stess

    if stress <= 50:
        print(str(days_left) + str(stress) + " = :)")
    elif stress >= 50 and stress <= 70:
        print(str(days_left) + str(stress) + " = :|")
    elif stress >= 70:
        print(str(days_left) + str(stress) + " = :(")

    if days_left <= 1:
        break

    time.sleep(2)


did_i_do_it = random.randint(0,1)


if did_i_do_it:
    print("Yayyy, success!")
else:
    print("cry.")
```

./Procrastination.py

```
910 = :)
820 = :)
730 = :)
640 = :)
550 = :)
460 = :|
370 = :|
280 = :(
190 = :(
Yayyy, success!
```

```
#-----------------------------------------------------------------
#
# Psycho
# Blake Irons
#
# Prints lines from the script to American Psycho
# that share a word the user selects.
#
#-----------------------------------------------------------------

import nltk


f=open('psycho2.txt','r')
raw=f.read()
tokens = nltk.word_tokenize(raw)
text = nltk.Text(tokens)


search = str(input('Enter a word to go psycho:\n'))
print('\n')


print('#'*84)
print('#' + ' '*82 + '#')


text.concordance(search,80, 2314)


print('#' + ' '*82 + '#')
print('#'*84)

print('\nMy punishment continues to elude me... and I gain no
deeper knowledge of myself. \nNo new knowledge can be extracted
from my telling. \nThis confession has meant... nothing. ')
```

```
Enter a word to go psycho:
love


##################################################################

 t mysterious little dish . '' You 'll love it . And then ... the
red snapper wit Sorbet ? JEAN Thanks , Patrick . I 'd love some .
Bateman walks in with a bottle many great ones , but `` The Greatest
Love of All '' is one of the best , most p

##################################################################

My punishment continues to elude me... and I gain no deeper knowl-
edge of myself.
No new knowledge can be extracted from my telling.
This confession has meant... nothing.



Enter a word to go psycho:
food


##################################################################

e welfare system . We have to provide food and shelter for the
homeless and oppo EMAN You want some money ? . Some ... food ? The
Homeless Man nods and starts to cry. Bateman reaches

##################################################################

My punishment continues to elude me... and I gain no deeper knowl-
edge of myself.
No new knowledge can be extracted from my telling.
This confession has meant... nothing.
```

```
#-----------------------------------------------------------------
#
# Shinning
# Yining Lai
#
#-----------------------------------------------------------------

from time import sleep
import sys

name = input ("Here is Overlook Hotel.
                 I am Jack Torrence, caretaker here.\
             \nI've been writing here for 5 months.
               You'd better not distract me.\
             \nWhat's your name?")
boy = {("boy"), ("man"), ("male")}
girl = {("girl"), ("woman"), ("female")}

while True:
    gender = input ("What's your gender?")
    if gender in boy:
       genderend = ("boy")
       break
    elif gender in girl:
        genderend = ("girl")
        break
    else:
       print ("What do you mean?")
       continue
writing = input ("Do you want to read my work?")
if writing != ("yes"):
   print ("Get the F out of here. Do not interrupt me again!")
   sys.exit()

print (".")
sleep(0.5)
print (".")
sleep(0.5)
print (".")
sleep(0.5)
waittime = 70
done = 0
while waittime is not done:
       print ("All work and no play makes" +
        " " + name + " " + "a dull" + " " + genderend)
       sleep(0.5)
       waittime -= 1
```

./Shinning.py

```
Here is Overlook Hotel. I am Jack Torrence, caretaker here.
I've been writing here for 5 months. You'd better not distract me.
What's your name?Leo
What's your gender?dog
What do you mean?
What's your gender?male
Do you want to read my work?yes



All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
```

./Shinning.py

```
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
All work and no play makes Leo a dull boy
```

```
#-----------------------------------------------------------------
#
# TempleOS: A Case Study
# Mary
#
#-----------------------------------------------------------------

import random

word_list = []

with open("file.txt") as file:
        for line in file:
                line = line.strip()
                word_list.append(line)

poem_str = ""

for x in range(0,5):
        poem_str += word_list[random.randint(0,len(word_list))] + " "

print()
print(poem_str)
print()
```

```
African
Angel
BBC
BRB
Bam
Boo
Burp
CIA
California
Catastrophic_Success
China
Church
Cosmos
Dad
Dudly_Doright
FBI
GarryKasparov
Ghost
Give_me_praise
God
God_is_not_mocked
God_smack
Greece
Greek_to_me
Han_shot_first
Hasta
Heaven
Hicc_up
HolySpirit
I'll_ask_nicely
I'll_be_back
I'll_get_right_on_it
I'll_let_you_know
I'll_think_about_it
I'm_God_and_you're_not
I'm_God_who_the_hell_are_
you
I'm_beginning_to_wonder
I'm_bored
I'm_busy
I'm_done
I'm_feeling_nice_today
I'm_gonna_smack_someone
I'm_good_you_good
I'm_grieved
I'm_impressed
I'm_in_suspense
I'm_not_dead_yet
I'm_not_sure
I'm_off_today
I'm_on_a_roll
I'm_the_boss
I'm_thrilled
I'm_tired_of_this
IMHO
I_am_not_amused
I_be_like
I_can't_believe_it
I_could_be_wrong
```

```
I_didn't_see_that
I_don't_care
I_donno
I_forgot
I_give_up
I_got_your_back
I_had_a_crazy_dream
I_hate_when_that_happens
I_have_an_idea
I_just_might
I_love_this
I_love_you
I_made_it_that_way
I_pity_the_fool
I_planned_that
I_quit
I_see_nothing
I_veto_that
I_was_just_thinking
I_was_sleeping
Icarus
If_had_my_druthers
Is_that_so
Is_that_your_final_answer
Isn't_that_special
It's_nice_being_God
It_grieves_me
Ivy_league
Japan
Jedi_mind_trick
Jesus
King_Midas
Knock_you_upside_the_head
LOL
Make_America_Great_Again
Mars
Mission_Accomplished
Mom
Moses
NOT
NeilDeGrasseTyson
Obama
Oh_Hell_No
Oh_really
Okilydokily
One_finger_salute
Oy
Pope
Putin
ROFLMAO
Russia
Shakespeare
Shalom
Shhh
StephenHawking
SupremerCourt
Terry
That's_gonna_leave_a_mark
That's_my_favorite
```

```
Varoom
Vegas
Venus
Watch_this
What
What_I_want
What_are_you_doing_Dave
WooHoo
Wow
Yawn
Yes_you_are
Yo
You_can_count_on_that
You_da_man
You_fix_it
You_get_what_you_pray_
for
You_know
Zap
Zzzzzzzz
a_flag_on_that_play
a_likely_story
a_screw_loose
abnormal
absotively_posilutely
absolutely
act
adjusted_for_inflation
adultery
after_a_break
ahh
ahh_thats_much_better
air_head
and_the_award_goes_to
and_then_what
angel
anger
application
are_you_deaf
are_you_feeling_lucky
are_you_insane
are_you_sure
arent_you_clever
arrogant
as_a_matter_of_fact
astounding
astronomical
astrophysics
atheist
atrocious
au_revoir
awesome
awful
ba_ha
bad
bad_ol_puddytat
baffling
bank
basically
```

./TempleOS: A Case Study.py

TempleOS is an operating system written by Terry A Davis.

He was a genius who wrote his wen coding language, editor, compiler, and kernel.

Unfortunately Terry was also schizophrenic and much of what he said was very problematic.

This was a case study on TempleOS, an operating system made with instructions from God designed solely to communicated with God.

Within the operating system, there are many games and files to explore.

One such file is entitles "HappyWords.txt".

This poem is a series of randomly selected words from such file, supposedly making a happy poem.

However, as you may see, most of these words have nothing to do with happiness.

Anyways, here is a poem, I guess by Terry A Davis technically, just compiled by me:

Oops Zzzzzzzz joy to_infinity_and_beyond mine

no_news_is_good_news how_could_you it'd_take_a_miracle
Han_shot_first do_you_like_it

white_trash act arent_you_clever I'm_good_you_good no_
news_is_good_news

```
battle                      debt                        food
be_happy                    delicious                   fool
be_quiet_bird               delightful                  fortitude
beam_me_up                  depressing                  foul
because_I_said_so           desert                      freak
beep_beep                   didn't_I_say_that           frown
begs_the_question           dignity                     fun
bickering                   do_I_have_to                funny
big_fish                    do_it                       furious
biggot                      do_not_disturb              gambling
birds                       do_over                     game_changer
bizarre                     do_you_get_a_cookie         game_over
blessing                    do_you_have_a_problem       geek
boink                       do_you_know_what_time_it_genius
boss                        is                          ghastly
break_some_woopass_on_you   do_you_like_it              ghetto
bring_it_on                 do_you_want_another         glam
bummer                      doh                         glorious
busybody                    don't_count_on_it           gluttony
but_of_course               don't_even_think_about_it   go_ahead_make_my_day
by_the_way                  don't_have_a_cow            good
bye                         don't_mention_it            gosh
can_you_hear_me_now         don't_push_it               gross
car                         don't_worry                 grumble
catastrophe                 downer                      guilty
caution                     drama                       guppy
chaos                       driving                     ha
charged                     duck_the_shoe               handyman
charity                     dude_such_a_scoffer         hang_in_there
check_this_out              earnest                     happy
cheerful                    economy                     happy_happy_joy_joy
chess                       eh                          hard_working
chill                       ehh_a_wise_guy              harder_than_it_looks
chill_out                   ehheh_that's_all_folks      hate
choose                      employee                    hateful
chump_change                employer                    he_be_like
church                      end                         heads_I_win_tails_you_
class__class__shutup        endeared                    lose
clever                      endurance                   heathen
climate                     endure                      hello
close_your_eyes             energy                      here_now
come_and_get_me             enough                      hey_Mikey_he_likes_it
comedy                      enthusiasm                  hey_thats_right
commanded                   envy                        hi
completely                  epic_fail                   high_five
computers                   et_tu                       high_mucky_muck
conservative                everything's_a_okay         hilarious
cosmetics                   evolution                   hippy
could_it_be_Satan           exorbitant                  hit
couldn't_be_better          experts                     ho_ho_ho
couldn't_possibly           exports                     hobnob
courage                     fabulous                    hold_on_a_minute
cowardice                   face_palm                   holier_than_thou
cracks_me_up                failure_is_not_an_option    holy_grail
crash_and_burn              failure_to_communicate      home
crazy                       fake                        homo
cursing                     fancy                       honestly
dance                       far_out_man                 honesty
```

```
#----------------------------------------------------------------
#
# the_rings
# Boyuan Zheng
#
#----------------------------------------------------------------

import time
import subprocess
import os
three = 3

ring = ""
for theElvenKings in "the sky":
    ring += ""
seven = 7
for theDwarfLords in "halls of stone":
    ring += ""
nine = 9
for mortalMen in "doomed to die":
    ring += ""
one = 1
for theDarkLord in "his dark throne":
    ring += ""
for whereTheShadowLie in " the land of mordor":
    ring += ""
one * ring > "to rule them all"
one * ring.find("them")
one * ring > "to bring them all and in the darkness bind them"
ring += "     O\n\n    "
for whereTheShadowLie in " the land of mordor":
    ring += "O"
    if whereTheShadowLie == "h" : ring += "\n\n  OO"
    if whereTheShadowLie == "n" : ring += "\n\n"

while 1:
    print ring.replace("I", "O")
    time.sleep(1)
    os.system("clear")

    print ring.replace("O", "I")
    time.sleep(1)
    os.system("clear")
    O           I           O           I           O
   OOO         III         OOO         III         OOO
 OOOOOOO     IIIIIII     OOOOOOO     IIIIIII     OOOOOOO
OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO

    I           O           I           O           I
   III         OOO         III         OOO         III
 IIIIIII     OOOOOOO     IIIIIII     OOOOOOO     IIIIIII
IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII

    O           I           O           I           O
   OOO         III         OOO         III         OOO
 OOOOOOO     IIIIIII     OOOOOOO     IIIIIII     OOOOOOO
OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO

    I           O           I           O           I
   III         OOO         III         OOO         III
 IIIIIII     OOOOOOO     IIIIIII     OOOOOOO     IIIIIII
IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII

    O           I           O           I           O
   OOO         III         OOO         III         OOO
 OOOOOOO     IIIIIII     OOOOOOO     IIIIIII     OOOOOOO
OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO

    I           O           I           O           I
   III         OOO         III         OOO         III
 IIIIIII     OOOOOOO     IIIIIII     OOOOOOO     IIIIIII
IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII

    O           I           O           I           O
   OOO         III         OOO         III         OOO
 OOOOOOO     IIIIIII     OOOOOOO     IIIIIII     OOOOOOO
OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO

    I           O           I           O           I
   III         OOO         III         OOO         III
 IIIIIII     OOOOOOO     IIIIIII     OOOOOOO     IIIIIII
IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII

    O           I           O           I           O
   OOO         III         OOO         III         OOO
 OOOOOOO     IIIIIII     OOOOOOO     IIIIIII     OOOOOOO
OOOOOOOOO   IIIIIIIII   OOOOOOOOO   IIIIIIIII   OOOOOOOOO
```

"... they are not bugs, they are features."

```
  ' _____ '       ' _____ '
 /  o o    \     /  o -    \
(    ^     ,)   (    ^     ,)
 \   0    /      \   7    /
  `._____.'       `._____.'
  /       \       /       \
 /         \     /         \
|           |   |           |
 \         /     \         /
  \       /       \       /
   -------         -------



  ' _____ '       ' _____ '
 /  - -    \     /  - -    \  z  z  Z  Z
(    ^     ,)   (    ^     ,)
 \   3    /      \   0    /
  `._____.'       `._____.'
  /       \       /       \
 /         \     /         \
|           |   |           |
 \         /     \         /
  \       /       \       /
   -------         -------
```