

```

# This function reads a text file and increments matrices accordingly.
def train_matrix(list,input_matrix,pitch_dict):
    #print list
    note1 = ''
    note2 = ''
    #print list
    for i in range(len(list) - 1):
        #print i
        note1 = define_note_value(list[i],pitch_dict)
        note2 = define_note_value(list[i+1],pitch_dict)
        #print note1
        #print note2
        input_matrix[note1][note2] += 1
    return input_matrix

# Converts pitch value of note into a proper MIDI value.
def convert_to_midi(note,pitch_dict):
    octave = 5
    note = str(note)
    midi_number = 12*(octave) + int(define_note_value(note,pitch_dict))
    return midi_number

# This function will find a random i value of the matrix that will serve as a starting note.
def find_start_note(trained_matrix,num_rows):
    valid_note = False
    while valid_note == False:
        starting_note = random.randrange(num_rows)
        for i in range(num_rows):
            if trained_matrix[starting_note][i] != 0:
                valid_note = True
                break
    return starting_note

# This function will figure out, based on the weights of the matrix, which notes to play.
def make_phrase(trained_matrix,starting_note,num_columns,phrase_length):
    phrase = []
    phrase.append(starting_note)
    possible_note_options = []
    next_note = 0
    max_weight = 0
    max_list = []
    current_phrase_length = 1
    while current_phrase_length < phrase_length:
        for i in range(num_columns):
            possible_note_options.append(trained_matrix[starting_note][i])
        max_weight = max(possible_note_options)
        #print max_weight
        for j in range(len(possible_note_options)):
            if max_weight == possible_note_options[j]:
                max_list.append(j)
        next_note = random.choice(max_list)
        phrase.append(next_note)
        starting_note = next_note
        current_phrase_length += 1
        possible_note_options = []
        max_list = []
    return phrase

```

0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0
0	0	0	0	0	3	0	2	0	0	0	0
0	0	0	0	0	0	0	0	0	5	0	0
0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	2	0
0	0	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	3	0	0	0	0	0	0	0	0	0

