RHYTHMIC RECURSION

CELEEN RUSK

HELLO@CELEEN.INFO

@CELEENR (TWITTER)

COMPUTER SCIENCE

COMPUTER ~~SCIENCE~~

~~COMPUTER SCIENCE~~

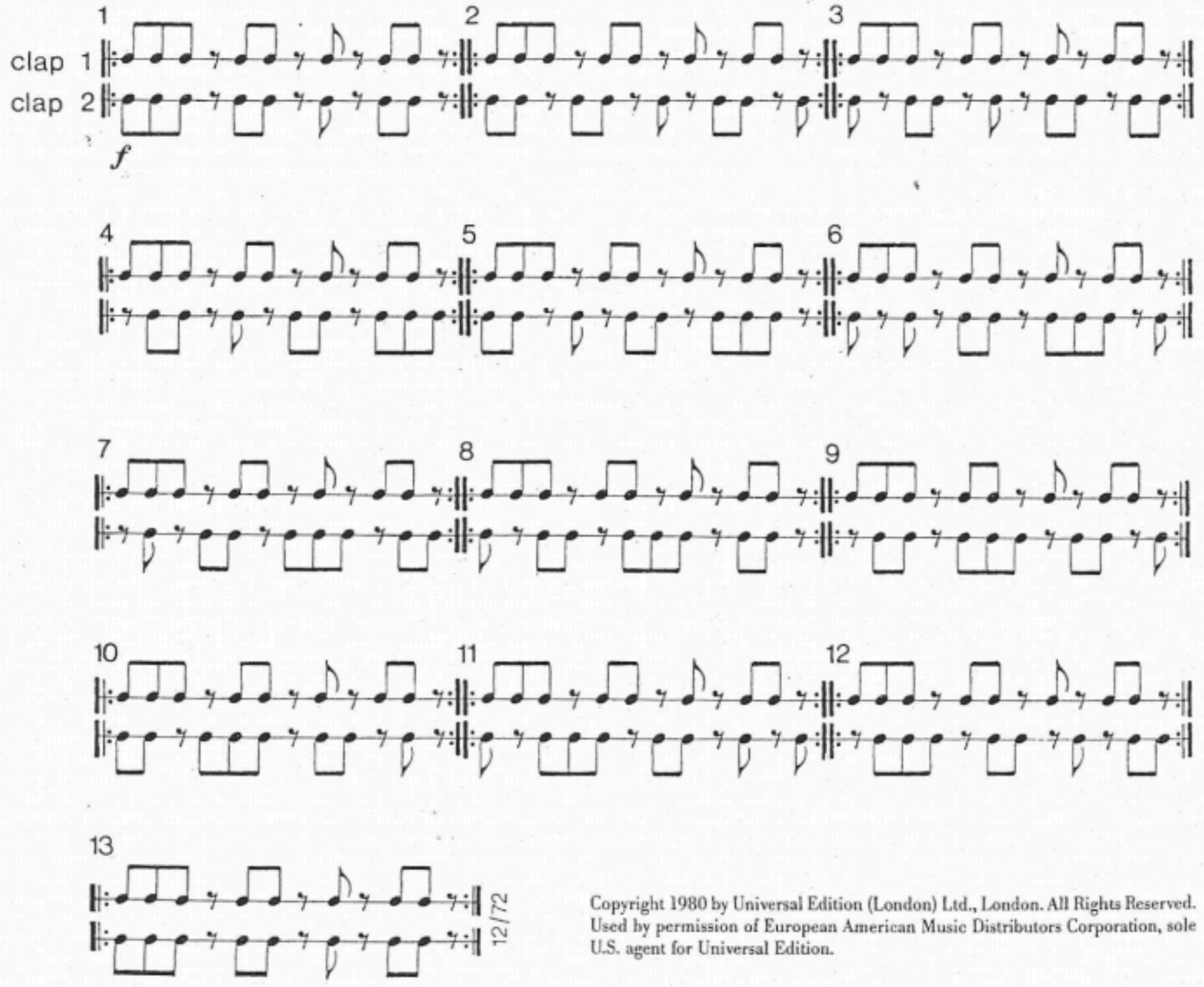
GEOMETRY

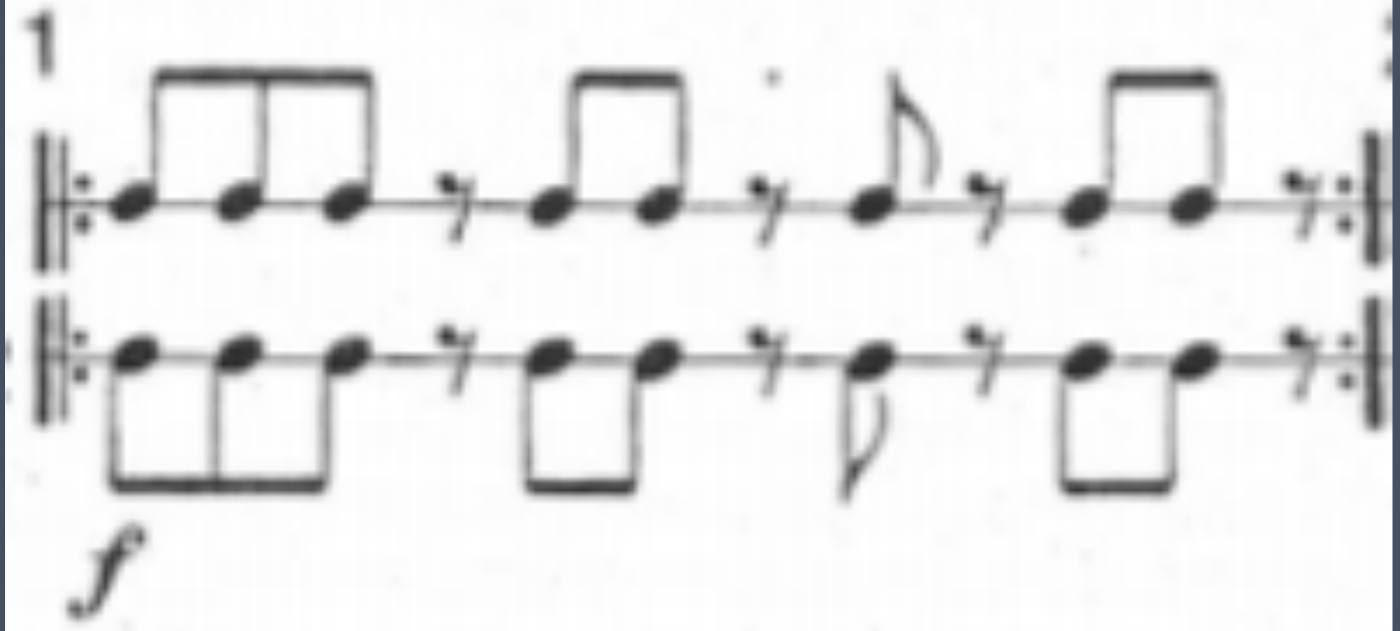
(GAIA) + (METRON)

COMPUTER SCIENCE IS THE STUDY OF **PROCESS**

1. REPRESENTATION
2. PROCESS
3. COMPOSITION

♩ = 160-184 Repeat each bar 12 times

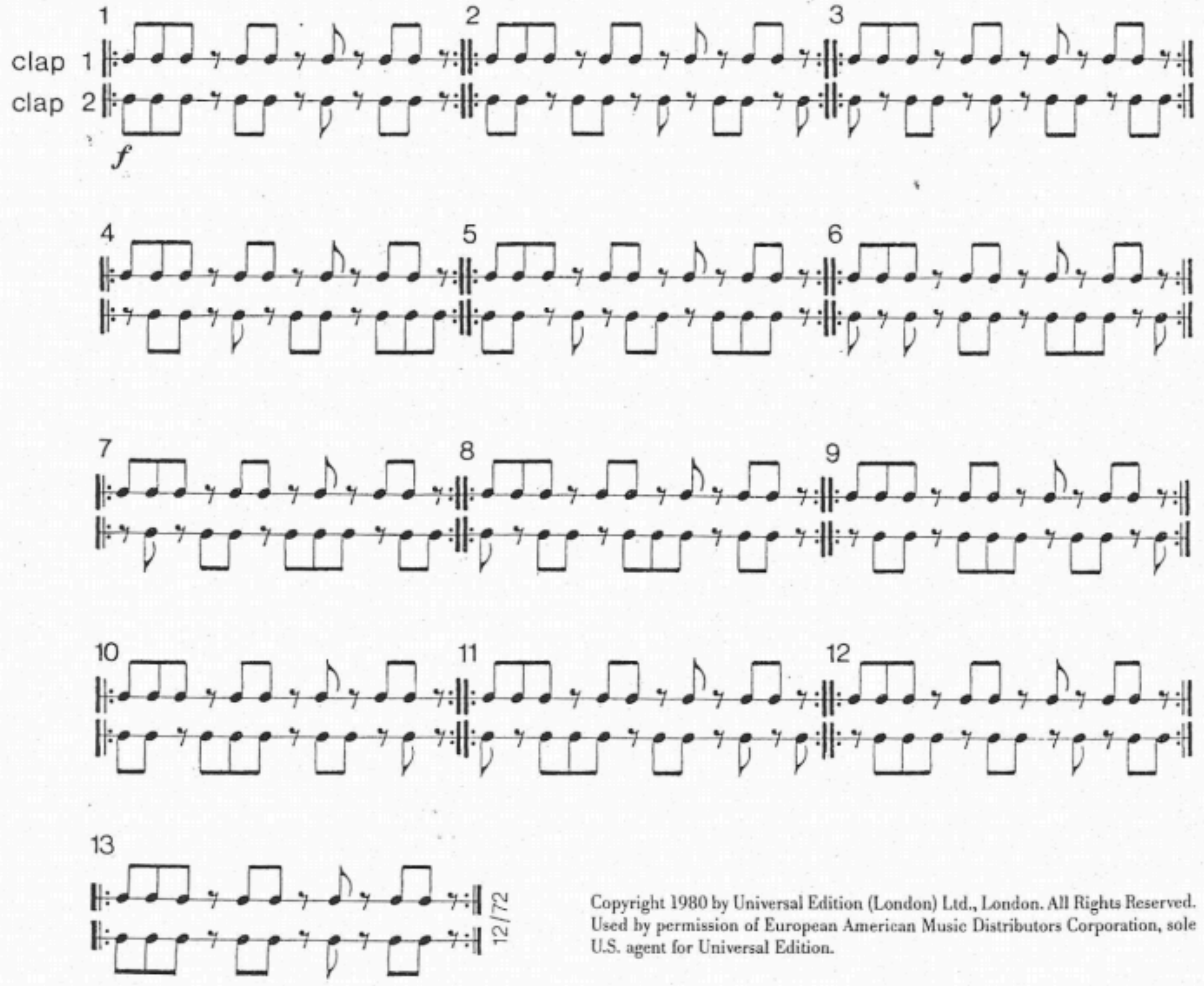



1
f

[1,1,1,0,1,1,0,1,0,1,1,0]

[1,1,1,0,1,1,0,1,0,1,1,0]

♩ = 160-184 Repeat each bar 12 times



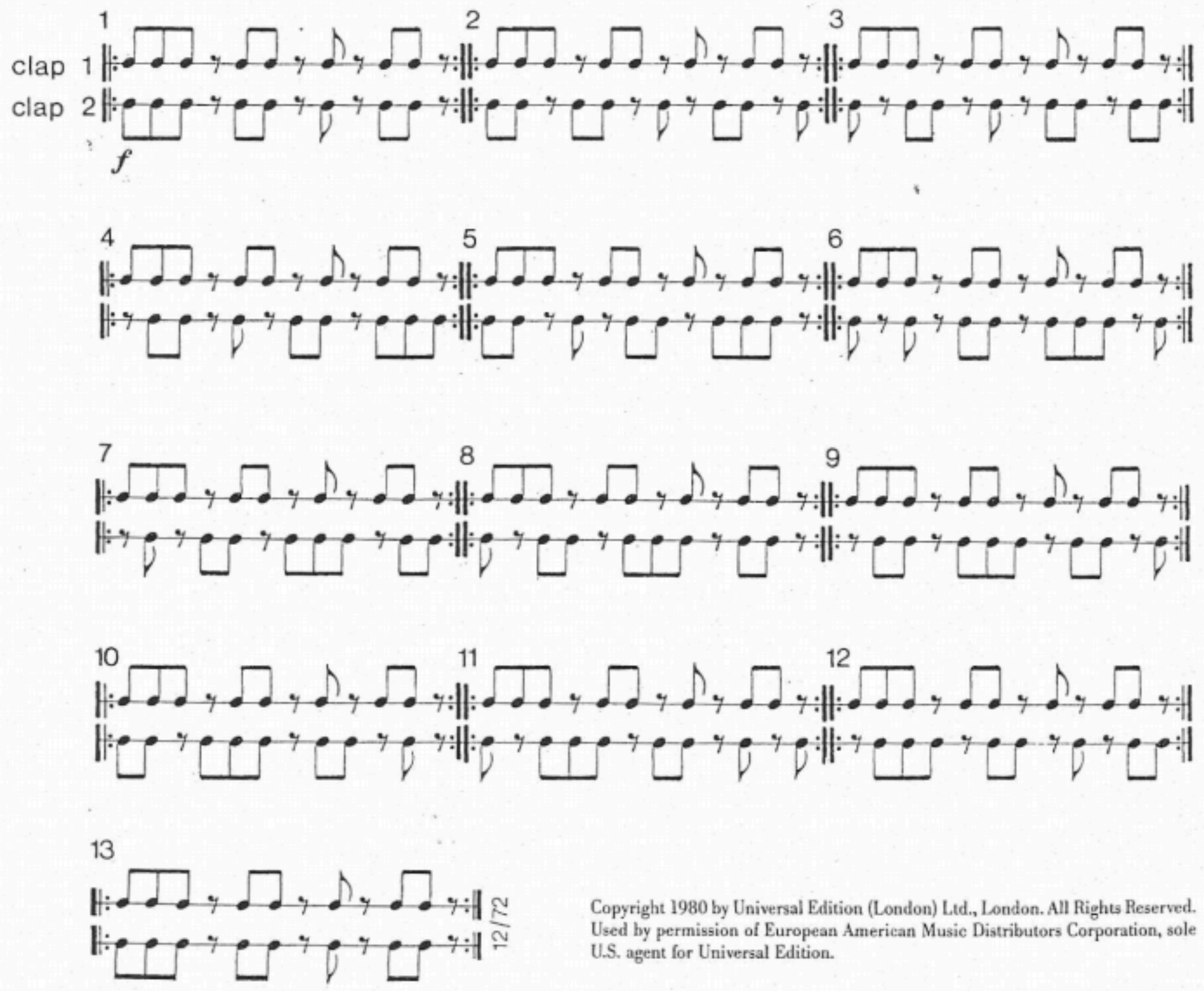
RECURSION : A METHOD OF DEFINING FUNCTIONS IN WHICH THE FUNCTION BEING DEFINED IS APPLIED WITHIN ITS OWN DEFINITION

--WIKIPEDIA

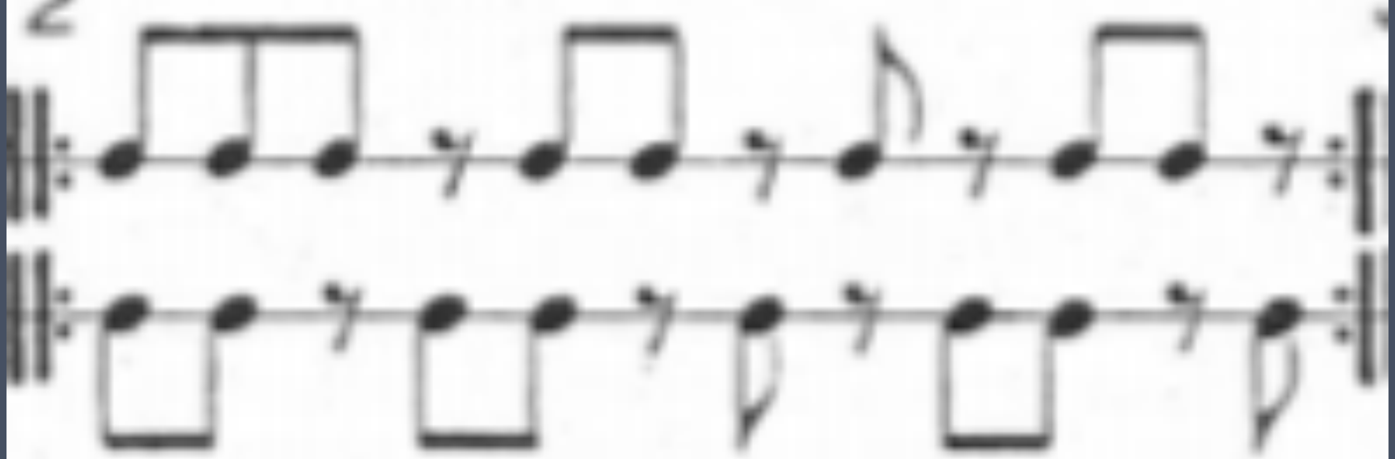
RECURSION : **(MATHEMATICS) AN EXPRESSION SUCH THAT EACH TERM IS GENERATED BY REPEATING A PARTICULAR (MATHEMATICAL) OPERATION**

--**WOLFRAM ALPHA**

♩ = 160-184 Repeat each bar 12 times

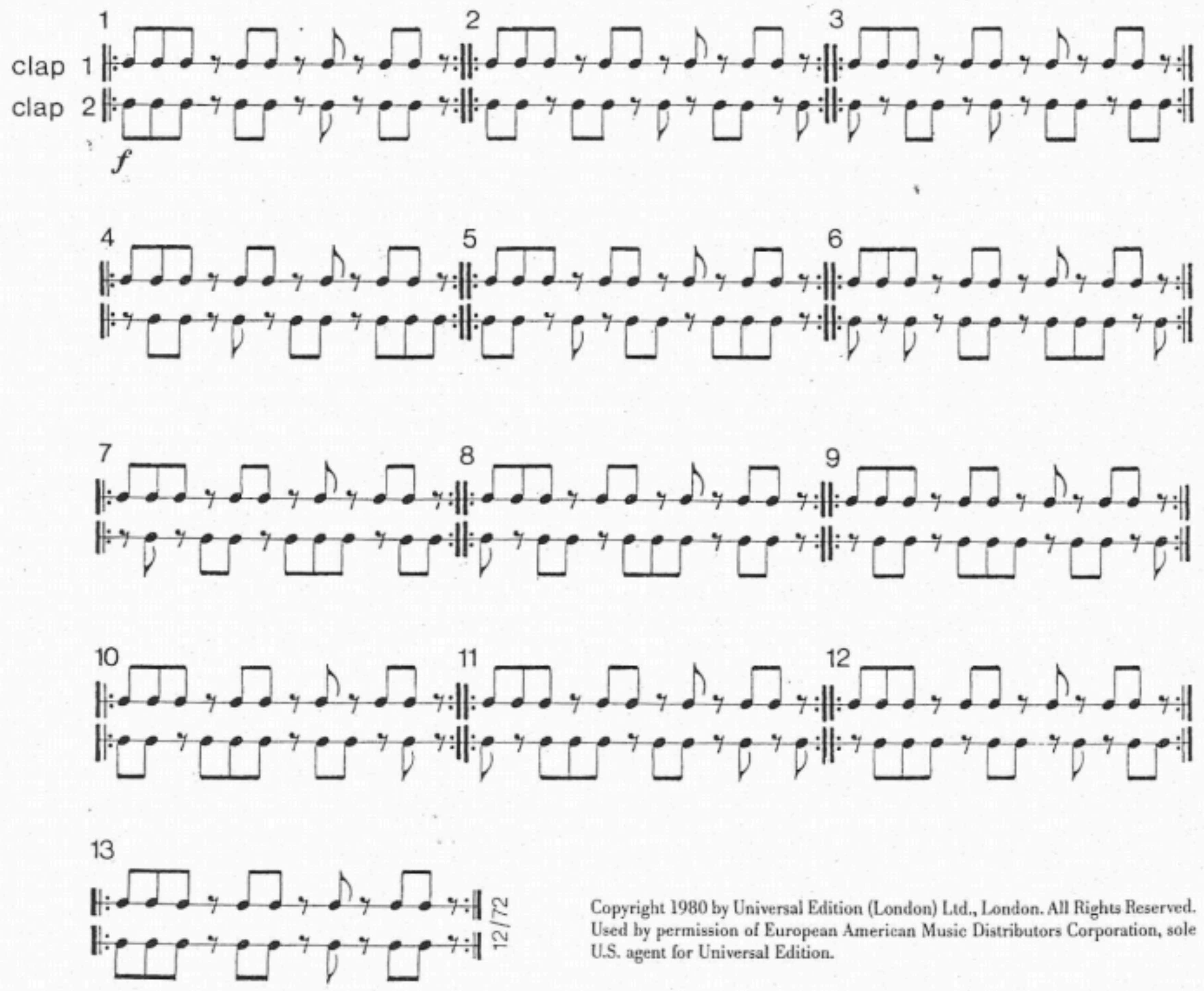


2



[1,1,1,0,1,1,0,1,0,1,1,0]
[1,1,0,1,1,0,1,0,1,1,0,1]

♩ = 160-184 Repeat each bar 12 times



AND NOW, IT'S TIME FOR THE BREAKDOWN

WHAT NEEDS REPRESENTATION?

1. **TWO PARTS**
   - PERFORMERS, OR VOICES
2. **WHAT THEY'RE PLAYING**
   - NOTES ON THE PAGE
3. **HOW TO PLAY THE PARTS**
   - THE PROCESS OF PLAYING OR PERFORMING THOSE NOTES

1. TWO PARTS

PERFORMERS, OR VOICES

```
@voice1 = :drum_tom_hi_hard
@voice2 = :drum_tom_mid_hard
```

2. WHAT THEY'RE PLAYING

NOTES ON THE PAGE

```
@baseline = [1,1,1,0,1,1,0,1,0,1,1,0]

rotating_part = @baseline.clone.rotate
=> [1,1,0,1,1,0,1,0,1,1,0,1]
```

3. HOW TO PLAY THE PARTS

THE PROCESS OF PLAYING OR PERFORMING THE NOTES

IMPORTANT TERMS

- MEASURE/BAR
- SECTION

THINGS THAT EXIST:

- `#play_note`
- `#play_rest`

OTHER HELPER METHODS

- #play_measure
- #play_section
- #both_parts_play_section

#play_measure

PLAYS THROUGH THE NOTES AND RESTS IN ONE 'MEASURE'

```ruby
def play_measure(pattern, voice)
  pattern.each do |value|
    value == 1 ? play_note(voice) : play_rest
  end
end

play_measure(@baseline, @voice1)
```

#play_section

REPEATS THE GIVEN MEASURE/PATTERN N TIMES

```
def play_section(pattern, voice)
  4.times do
    play_measure(pattern, voice)
  end
end

play_section(@baseline, @voice1)
```

#both_parts_play_section

BOTH PARTS PLAY THROUGH ONE SECTION EACH, SIMULTANEOUSLY

```
def both_parts_play_section(part2)
  in_thread do
    play_section(@baseline, @voice1)
  end

  play_section(part2, @voice2)
end

both_parts_play_section(@baseline)
```

RECURSION!

RECURSION!

```
def play_recursive_bit(rotating_part)
  play_recursive_bit(???)
end
```

RECURSION!

```ruby
def play_recursive_bit(rotating_part)
  return if rotating_part == @baseline

  play_recursive_bit(???)
end
```

RECURSION!

```
def play_recursive_bit(rotating_part)
  return if rotating_part == @baseline

  play_recursive_bit(rotating_part.rotate)
end
```

RECURSION!

```
def play_recursive_bit(rotating_part)
  both_parts_play_section(rotating_part)

  return if rotating_part == @baseline

  play_recursive_bit(rotating_part.rotate)
end
```

ALL TOGETHER, NOW

```ruby
def play_piece
  play_recursive_bit(@baseline.rotate)
end
```

ALL TOGETHER, NOW

```
def play_piece
  both_parts_play_section(@baseline)
  play_recursive_bit(@baseline.rotate)
end

play_piece
```

RESOURCES AND LINKS

- **SONIC PI:** HTTP://SONIC-PI.NET/
- **THE COMPLETED CODE:** HTTPS://GITHUB.COM/CELEEN/CLAPPING_MUSIC/BLOB/MASTER/CLAPPING_MUSIC.RB
- **MY SLIDES:** HTTPS://GITHUB.COM/CELEEN/CELEEN.INFO/BLOB/MASTER/SOURCE/DECKSETS/RHYTHMIC_RECURSION.PDF

THANKS TO

- DOWNEY AND JOSH
- DBC
- SAM AARON, AND EVERYONE WHO CONTRIBUTES TO SONIC PI

CELEEN RUSK

HELLO@CELEEN.INFO

@CELEENR (TWITTER)