

## Second Racket Programming Assignment Solution

---

### Learning Abstract:

In this Racket assignment I learned about a variety of functions within the 2htdp/image library that allowed me to produce different visual programs such as Permutations of Disks (Task 1), Hirst Dots (Task 3), Stella Thing (Task 4), and a visual creation of my own (Task 5). Also within this assignment, the concept of recursion was used throughout the tasks below, except for Task 1. Along with visual programs, Task 2 explored numeric sequences.

### Task 1 - Permutations of Randomly Colored Stacked Dots

---

Code for Permutations of Randomly Colored Stacked Dots:

```
( require 2htdp/image )

( define ( tile sqre c1 c2 c3 )
  ( define the-square ( square 100 "solid" sqre ))
  ( define disk1 ( circle 45 "solid" c1 ))
  ( define disk2 ( circle 30 "solid" c2 ))
  ( define disk3 ( circle 15 "solid" c3 ))
  ( overlay disk3 disk2 disk1 the-square )
)

( define ( dots-permutations c1 c2 c3 )
  ( beside
    ( tile "black" c1 c2 c3 ) ( tile "black" c1 c3 c2 )
    ( tile "black" c2 c1 c3 ) ( tile "black" c2 c3 c1 )
    ( tile "black" c3 c1 c2 ) ( tile "black" c3 c2 c1 )
  )
)
```

Demo for Permutations of Randomly Colored Stacked Dots:

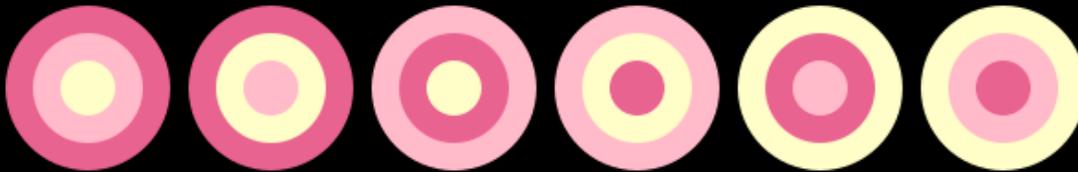
```
> ( tile "Dark Salmon" "Peach Puff" "Papaya Whip" "Light Salmon" )
```



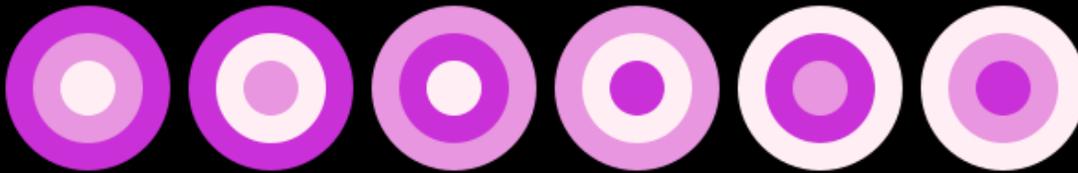
```
> ( tile "Olive" "Dark Khaki" "Pale Goldenrod" "Tan" )
```



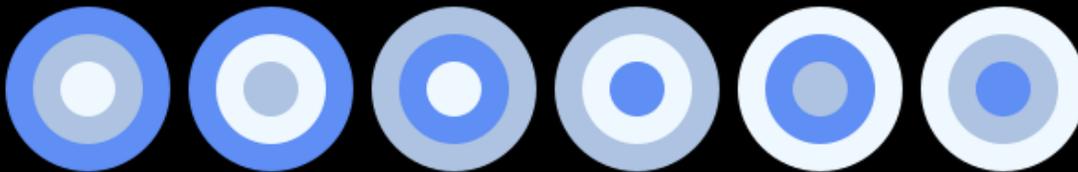
```
> ( dots-permutations "Pale Violet Red" "Pink" "Lemon Chiffon" )
```



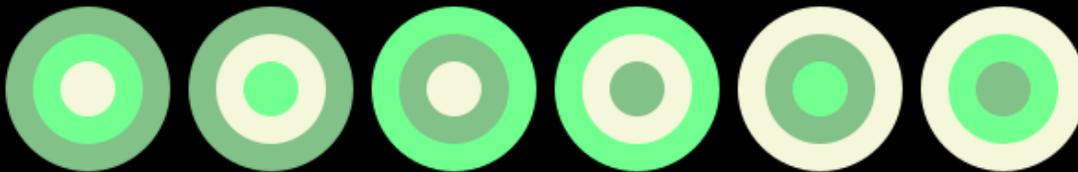
```
> ( dots-permutations "Medium Orchid" "Plum" "Lavender Blush" )
```



```
> ( dots-permutations "CornflowerBlue" "Light Steel Blue" "Alice Blue" )
```



```
> ( dots-permutations "Dark Sea Green" "Pale Green" "Beige" )
```



## Task 2 - Number Sequences

---

Code for Number Sequences :

```
#lang racket

( define ( natural-sequence n )
  ( cond
    ( ( > n 0 )
      ( natural-sequence ( - n 1 ) )
      ( display n ) ( display " " )
    )
  )
)

( define ( copies input n )
  ( cond
    ( ( > n 0 )
      ( copies input ( - n 1 ) )
      ( display input ) ( display " " )
    )
  )
)

( define ( special-natural-sequence n )
  ( cond
    ( ( > n 0 )
      ( special-natural-sequence ( - n 1 ) )
      ( copies n ( - n 1 ) )
      ( display n ) ( display " " )
    )
  )
)
```



### Task 3 - Hirst Dots

---

Code for Hirst Dots:

```
( require 2htdp/image )
( define ( random-color ) ( color ( random 256 ) ( random 256 ) ( random 256 ) ) )

( define ( tile )
  ( define the-square ( square 50 "solid" "Black" ) )
  ( define random-color-circle ( circle 15 "solid" ( random-color ) ) )
  ( overlay random-color-circle the-square )
)

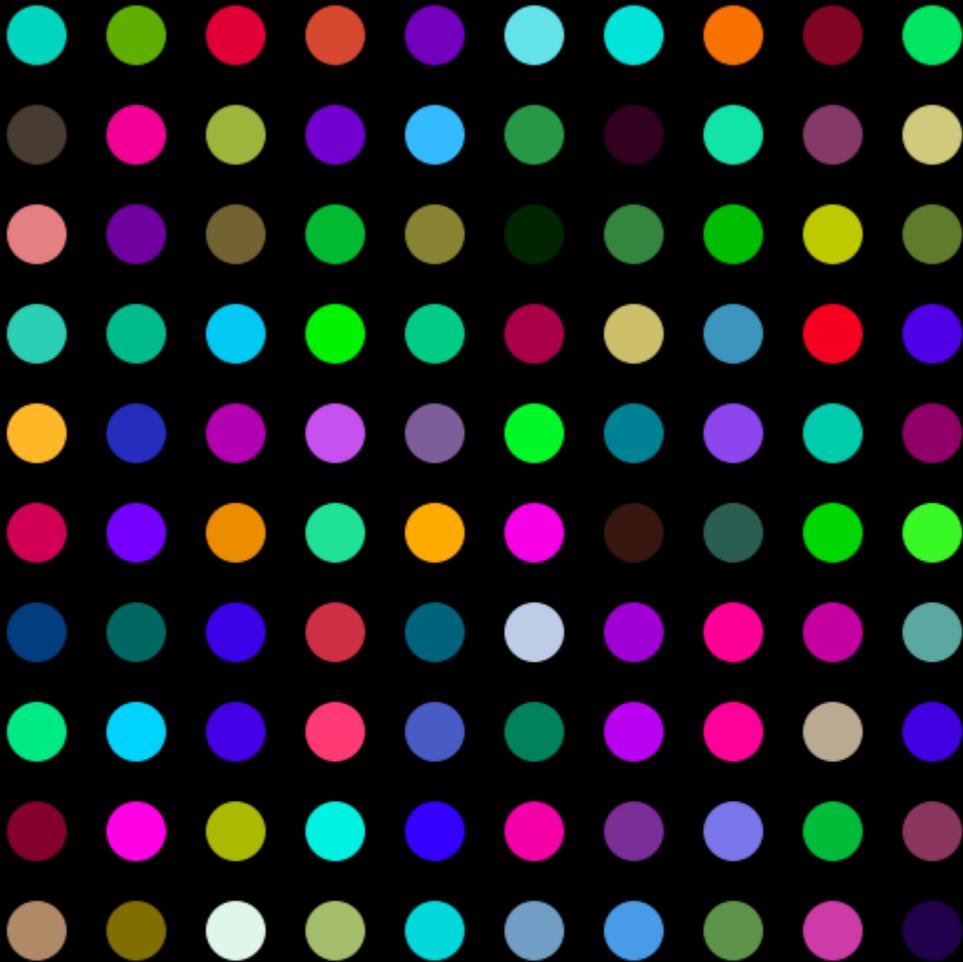
( define ( row-of-dots n )
  ( cond
    ( ( = n 0 )
      empty-image
    )
    ( ( > n 0 )
      ( beside ( row-of-dots ( - n 1 ) )
                ( tile ) )
    )
  )
)

( define ( rectangle-of-dots r c )
  ( cond
    ( ( = r 0 )
      empty-image
    )
    ( ( > r 0 )
      ( above ( rectangle-of-dots ( - r 1 ) c ) ( row-of-dots c ) )
    )
  )
)

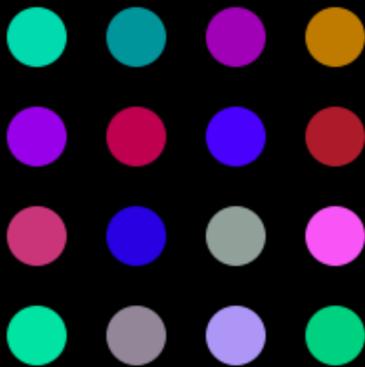
( define ( hirst-dots n )
  ( rectangle-of-dots n n )
)
```

Demo for Hirst Dots:

```
> ( hirst-dots 10 )
```



```
> ( hirst-dots 4 )
```



## Task 4 - Stella Thing

---

Code for Stella Thing:

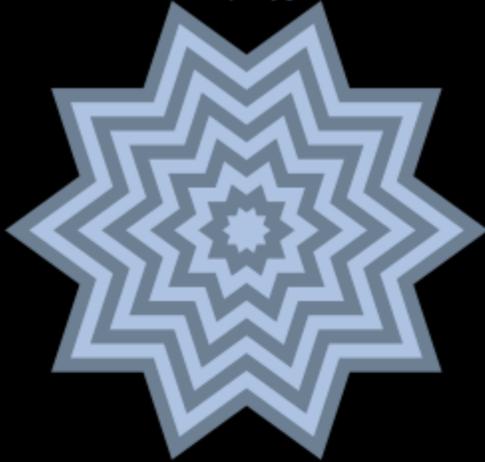
```
( require 2htdp/image )

( define ( nested-star-polygon side count c1 c2 )
  ( define spacing ( / side count ) )
  ( paint-nested-star-polygon 1 count spacing c1 c2 )
)

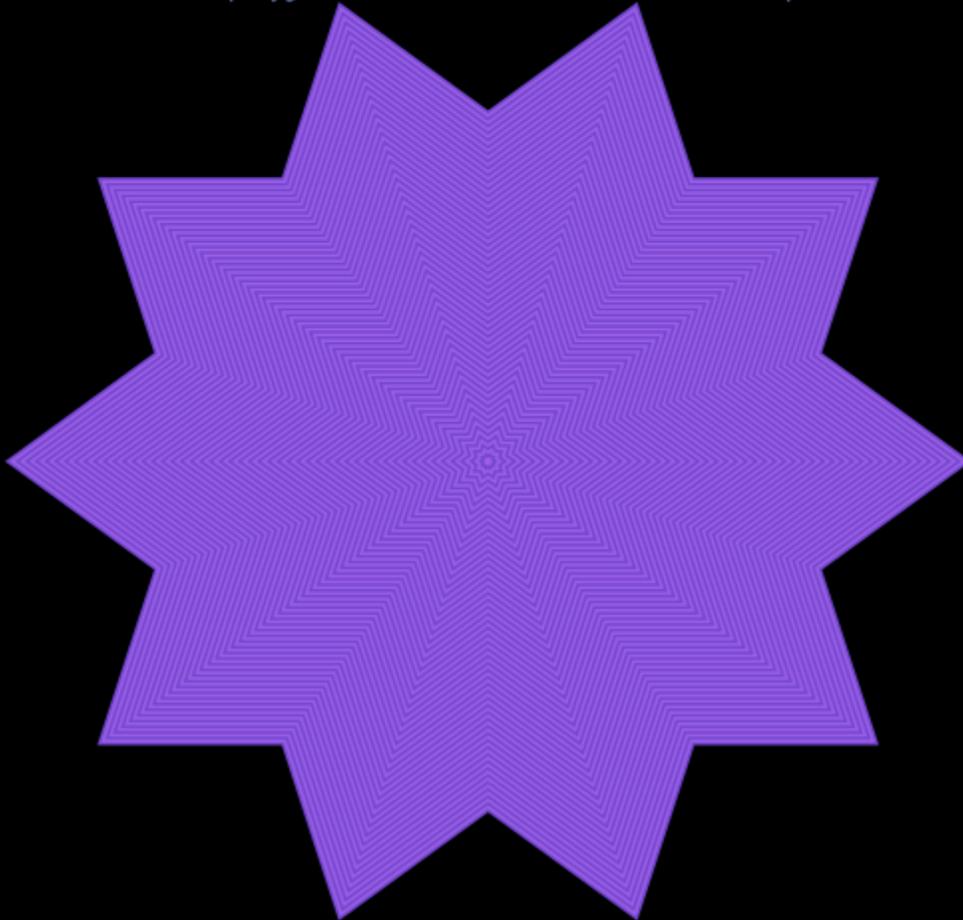
( define ( paint-nested-star-polygon from to spacing c1 c2 )
  ( define side-length ( * from spacing ) )
  ( define side-count 10 )
  ( define step-count 3 )
  ( cond
    ( ( = from to )
      ( if ( even? from )
        ( star-polygon side-length side-count step-count "solid" c1 )
        ( star-polygon side-length side-count step-count "solid" c2 )
      )
    )
    ( ( < from to )
      ( if ( even? from )
        ( overlay
          ( star-polygon side-length side-count step-count "solid" c1 )
          ( paint-nested-star-polygon ( + from 1 ) to spacing c1 c2 )
        )
        ( overlay
          ( star-polygon side-length side-count step-count "solid" c2 )
          ( paint-nested-star-polygon ( + from 1 ) to spacing c1 c2 )
        )
      )
    )
  )
)
)
```

Demo for Stella Thing:

```
> ( nested-star-polygon 75 10 "Slate Gray" "Light Steel Blue" )
```



```
> ( nested-star-polygon 150 100 "Slate Blue" "Medium Purple" )
```



## Task 5 - Creation

---

### Code for Creation:

```
( require 2htdp/image )
( define ( random-color ) ( color ( random 256 ) ( random 256 ) ( random 256 ) ) )

( define ( pac-man )
  ( underlay/offset ( rotate 30 ( wedge 50 300 "solid" "Yellow" ) )
    -20 -10
    ( circle 10 "solid" "Black" ) ) )

( define ( row-of-candy n )
  ( cond
    ( ( = n 0 )
      empty-image
    )
    ( ( > n 0 )
      ( beside ( row-of-candy ( - n 1 ) )
        ( overlay ( underlay
          ( ellipse 10 60 40 ( random-color ) )
          ( ellipse 20 50 40 ( random-color ) )
          ( ellipse 30 40 40 ( random-color ) )
          ( ellipse 40 30 40 ( random-color ) )
          ( ellipse 50 20 40 ( random-color ) )
          ( ellipse 60 10 40 ( random-color ) ) ) ( rectangle 100 20 "solid" "Black" ) ) )
    )
  )
)

( define ( my-creation n )
  ( beside ( pac-man ) ( row-of-candy n ) ) )
```

### Demo for Creation:

