Final Problem: 116

Abby Bernhardt

25 April, 2019

I went about solving this by going through the problem criteria given and making sure I fully understood what was being asked. Throughout the semester I have found this the easiest way to try to walk through these difficult problems. I knew I needed a variable N, so I started with that, and then I defined the colors, red, green and blue with the numbers given. From there I knew I needed some sort of definition for the new colored tiles, and a for loop to go through all of the possible iterations.

One of the only patterns I found was when you increased the number of red, green and blue tiles, but left the N alone, there were not a lot of ways to place all the colored tiles separately on the N blank squares. However, if you increase them both, the number get significantly larger, which makes sense because there are far more number of colors and spaces for them to fill. Lastly, as you only increase the N, the number outputted gets significantly larger by every 1 to 5 increment. For example, when N=10 there are 128 separate ways to place the colors on the tile, however, when N = 15, there are over 1400. This is not much of a pattern because every new input definitely increases, but does not increase by a constant amount of separate ways each time.

One question I think would be interesting to investigate would be adding more colors and seeing how they effect the numbers.

```
Function: tiles (NewColor, N)
Input: N = The number changed in the problem and represents
the number of units of black tiles.
Output: Number of different ways to place the colored tiles
separately in N blank black tiles.

N = 20 # this is the number we change
red = 2
green = 3
blue = 4

def tiles (NewColor, N):
# NewColor = number of black tiles covered
by a new color and N is total black tiles
    number = [1] * NewColor + [0] * (N-NewColor+1)
```

```
#number is the total number of ways which the
NewColors block can cover N black coloured tiles


    for ii in range (NewColor, N+1):
created a for loop to repeat from the number of
black tiles covered by a new color to the original N + 1.

All = tiles (red, N) + tiles( green, N) + tiles (blue, N)
This variable adds the separate numbers of red, green and
blue tiles together, to get the total number and the
number that is the purpose of this code.

There are two print statements. The first prints the
All variable just defined above which again shows the
total number of each color separately added together.
The second print statement individually shows the number
of red tiles it takes to fill N, the number of green
tiles it takes to fill N and the number of blue tiles
it takes to fill N.
```