Scrap code for warmup

This is the first MATH TEACHERS CODE Session

In [1]:

```
for i in 1:3
    println("Hello everyone, welcome to MATH TEACHERS CODE!")
end
```

```
Hello everyone, welcome to MATH TEACHERS CODE!
Hello everyone, welcome to MATH TEACHERS CODE!
Hello everyone, welcome to MATH TEACHERS CODE!
```

In [2]:

```
1+1
```

Out[2]:

```
2
```

In [3]:

```
x = 1
```

Out[3]:

```
1
```

In [4]:

```
x+1
```

Out[4]:

```
2
```

Getting to business: The Hailstone Sequence

In [25]:

```
isEven(x) = x % 2 == 0
next(x) = isEven(x) ? Int(x/2) : 3x+1
```

Out[25]:

```
next (generic function with 1 method)
```

In [29]:

```
x = 7
println(x)
x = next(x)
println(x)
x = next(x)
println(x)
x = next(x)
println(x)
```

```
7
22
11
34
```

Loop attempt 1

In [38]:

```
x = 7
for _ in 1:12
    print(x," ")
    x = next(x)
end
```

```
7 22 11 34 17 52 26 13 40 20 10 5
```

A more suitable loop

In [46]:

```
function printHail(x)
    while x != 1
        print(x," ")
        x = next(x)
    end
    print(x," ")
end
```

Out[46]:

```
printHail (generic function with 1 method)
```

In [49]:

```
printHail(323523559)
```

```
323523559 970570678 485285339 1455856018 727928009 2183784028 10918920
14 545946007 1637838022 818919011 2456757034 1228378517 3685135552 184
2567776 921283888 460641944 230320972 115160486 57580243 172740730 863
70365 259111096 129555548 64777774 32388887 97166662 48583331 14574999
4 72874997 218624992 109312496 54656248 27328124 13664062 6832031 2049
6094 10248047 30744142 15372071 46116214 23058107 69174322 34587161 10
3761484 51880742 25940371 77821114 38910557 116731672 58365836 2918291
8 14591459 43774378 21887189 65661568 32830784 16415392 8207696 410384
8 2051924 1025962 512981 1538944 769472 384736 192368 96184 48092 2404
6 12023 36070 18035 54106 27053 81160 40580 20290 10145 30436 15218 76
09 22828 11414 5707 17122 8561 25684 12842 6421 19264 9632 4816 2408 1
204 602 301 904 452 226 113 340 170 85 256 128 64 32 16 8 4 2 1
```

In [50]:

```
myArray = [2,4,5,62]
```

Out[50]:

```
4-element Array{Int64,1}:
  2
  4
  5
 62
```

In [51]:

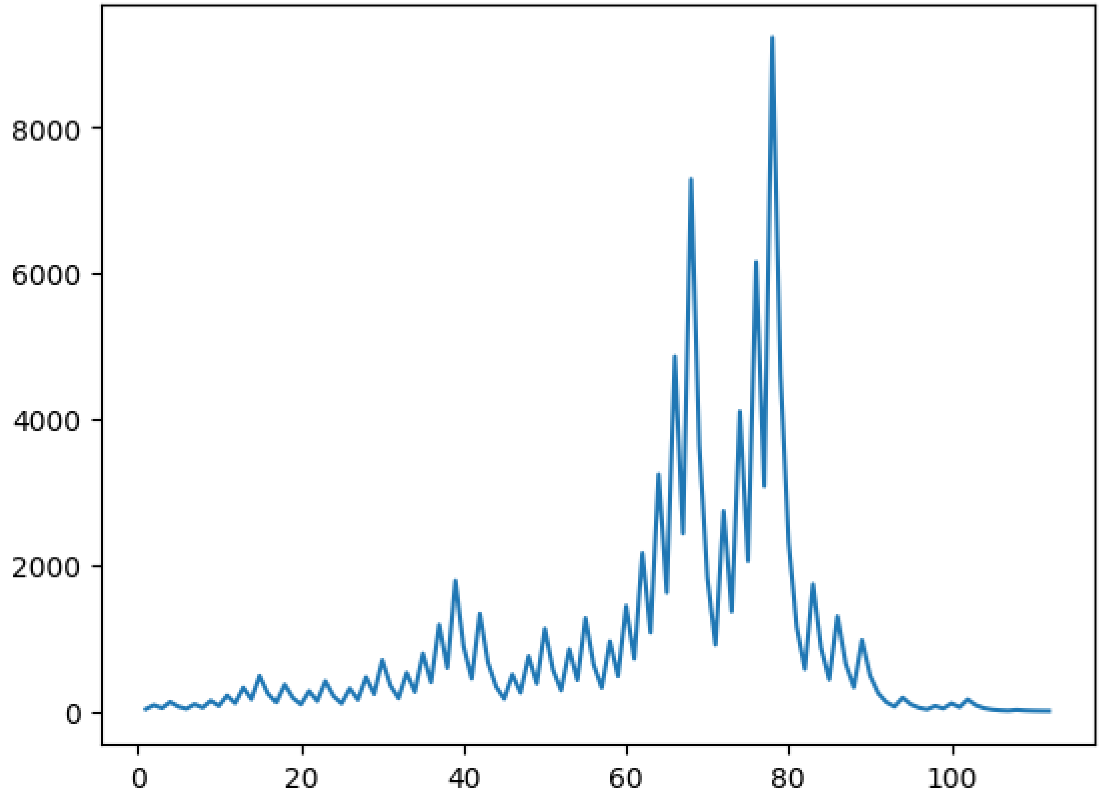
```
function hailArray(x)
    arrOut = [x]
    while x != 1
        x = next(x)
        push!(arrOut,x)
    end
    return arrOut
end
```

Out[51]:

```
hailArray (generic function with 1 method)
```

In [61]:

```
using PyPlot
start = 27
out = hailArray(start)
n = length(out)
plot(1:n,out)
```



Out[61]:

```
1-element Array{PyCall.PyObject,1}:
 PyObject <matplotlib.lines.Line2D object at 0x7f03642ccbd0>
```

In [62]:

```
? push!
```

```
search: push! pushfirst! pushdisplay
```

Out[62]:

```
  push!(collection, items...) -> collection

  Insert one or more items at the end of collection.

  Examples
  ==========

  julia> push!([1, 2, 3], 4, 5, 6)
  6-element Array{Int64,1}:
   1
   2
   3
   4
   5
   6

  Use append! to add all the elements of another collection to collect
ion. The
  result of the preceding example is equivalent to append!([1, 2, 3],
 [4, 5,
  6]).
```

In [ ]:

```

```