

In [1]:

```
using LinearAlgebra
```

In [2]:

```
x=rand(20)
```

Out[2]:

20-element Array{Float64,1}:

```
0.9608903686933943  
0.39401774336639406  
0.08658131845615324  
0.7663034051964499  
0.7784927683634106  
0.23350764006706215  
0.21884371484151433  
0.42808136279993936  
0.8362421772350752  
0.7972956260004989  
0.48330856510056575  
0.7452736937578635  
0.2451939655235551  
0.6243324771524765  
0.018295063790823463  
0.0485836223124696  
0.18692859990284072  
0.8564527128692809  
0.5404834891061703  
0.3597556746987838
```

In [3]:

```
y=x
for i=1:20
    y[i]=100*x[i]
end
return y
```

Out[3]:

```
20-element Array{Float64,1}:
 96.08903686933942
 39.401774336639406
  8.658131845615324
 76.63034051964499
 77.84927683634106
 23.350764006706214
 21.884371484151433
 42.80813627999394
 83.62421772350753
 79.72956260004989
 48.33085651005658
 74.52736937578635
 24.51939655235551
 62.43324771524765
  1.8295063790823463
  4.85836223124696
 18.69285999028407
 85.6452712869281
 54.04834891061703
 35.97556746987838
```

In [4]:

```
for i=1:20
    y[i]=floor(y[i])
end
return y
```

Out[4]:

20-element Array{Float64,1}:

```
96.0
39.0
 8.0
76.0
77.0
23.0
21.0
42.0
83.0
79.0
48.0
74.0
24.0
62.0
 1.0
 4.0
18.0
85.0
54.0
35.0
```

In [5]:

```
sort!(y)
```

Out[5]:

20-element Array{Float64,1}:

```
 1.0
 4.0
 8.0
18.0
21.0
23.0
24.0
35.0
39.0
42.0
48.0
54.0
62.0
74.0
76.0
77.0
79.0
83.0
85.0
96.0
```

In [1]:

```
(47/60)
```

Out[1]:

```
0.7833333333333333
```

In [9]:

```
PA(a,r,n)=a+r*(n-1)
```

Out[9]:

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

In [13]:

```
PA(11,7,5)
```

Out[13]:

```
39
```

In [32]:

```
x=zeros(20)
```

Out[32]:

```
20-element Array{Float64,1}:
```

```
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0  
0.0
```

In [33]:

```
for i=1:20
    x[i]=PA(11,7,i)
end
x
```

Out[33]:

20-element Array{Float64,1}:

```
11.0
18.0
25.0
32.0
39.0
46.0
53.0
60.0
67.0
74.0
81.0
88.0
95.0
102.0
109.0
116.0
123.0
130.0
137.0
144.0
```

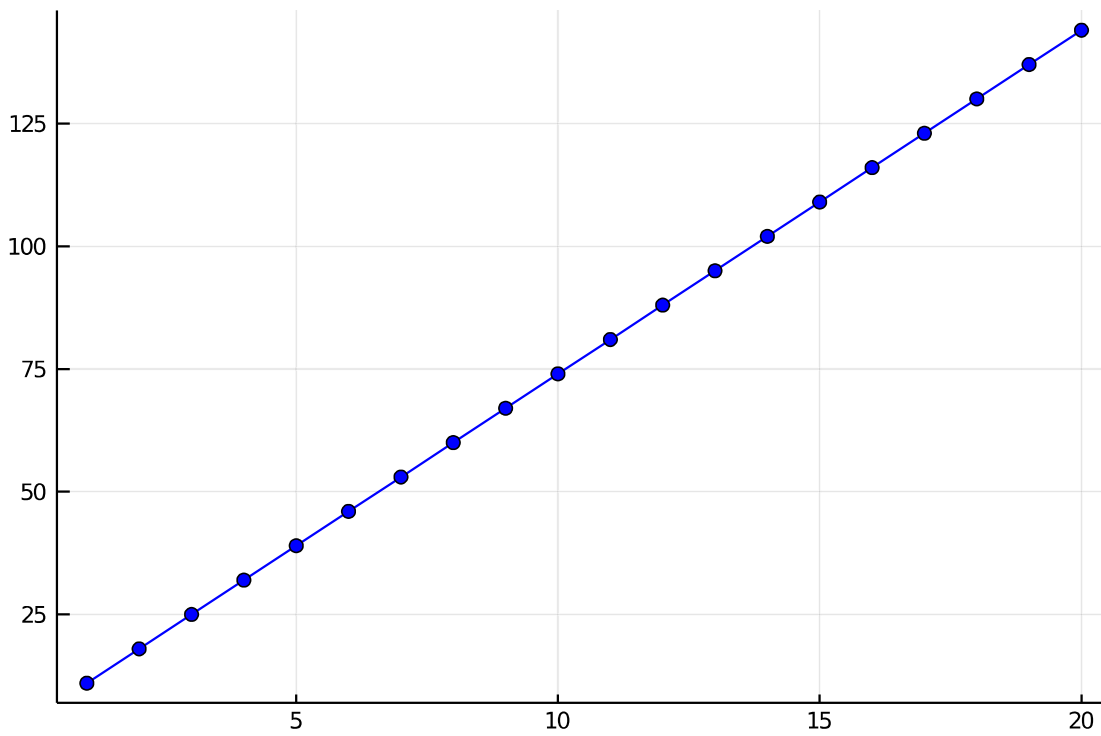
In [17]:

```
using Plots
```

In [34]:

```
plot(x,marker=:circle, color=:blue, leg=false)
```

Out[34]:



In [35]:

```
PG(a,r,n)=a*(r^(n-1))
```

Out[35]:

PG (generic function with 1 method)

In [36]:

```
PG(2,2,4)
```

Out[36]:

16

In [39]:

```
y=zeros(20)
for i=1:20
    y[i]=PG(3,1.7,i)
end
y
```

Out[39]:

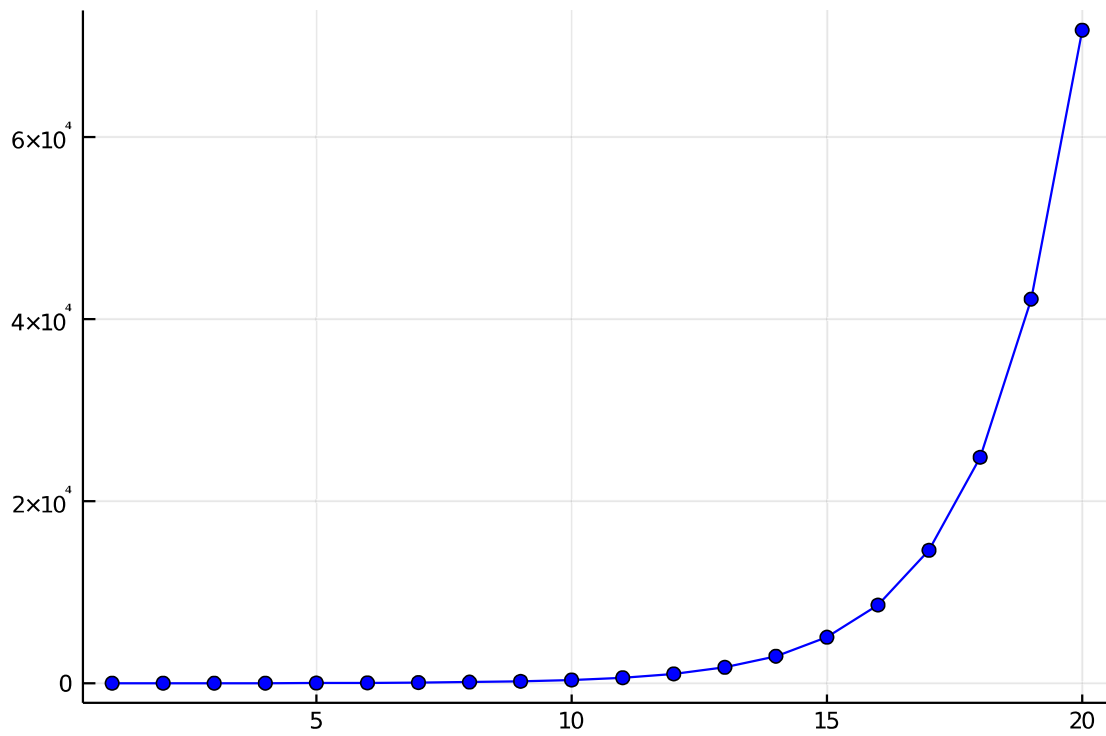
20-element Array{Float64,1}:

```
 3.0
 5.1
 8.669999999999998
14.738999999999997
25.056299999999993
42.59571
72.412706999999998
123.101601899999996
209.272723229999997
355.76362949099999
604.7981701346998
1028.1568892289897
1747.8667116892825
2971.37340987178
5051.334796782026
8587.269154529444
14598.357562700054
24817.20785659009
42189.25335620315
71721.73070554537
```

In [40]:

```
plot(y,marker=:circle, color=:blue, leg=false)
```

Out[40]:



In [29]:

```
function fib(n)
  if n==0
    return 0
  elseif n==1
    return 1
  else return fib(n-1)+fib(n-2)
  end
end
```

Out[29]:

fib (generic function with 1 method)

In [30]:

```
z=zeros(20)
for i=1:20
  z[i]=fib(i)
end
z
```

Out[30]:

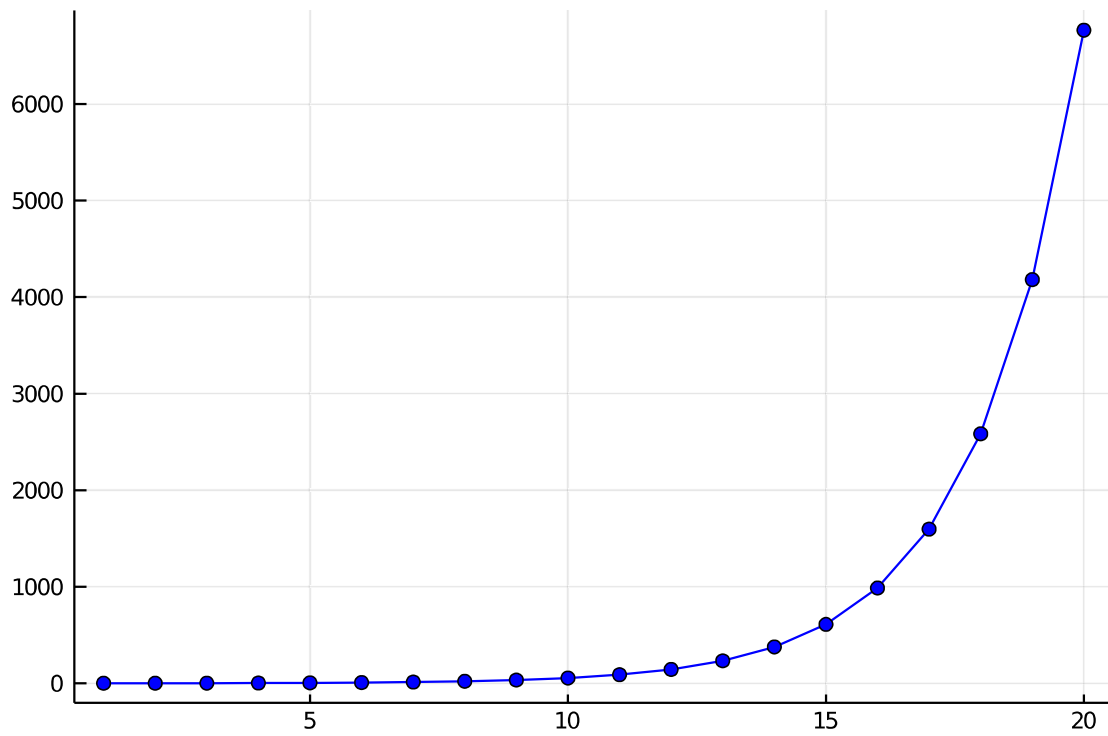
20-element Array{Float64,1}:

```
 1.0
 1.0
 2.0
 3.0
 5.0
 8.0
13.0
21.0
34.0
55.0
89.0
144.0
233.0
377.0
610.0
987.0
1597.0
2584.0
4181.0
6765.0
```

In [31]:

```
plot(z,marker=:circle, color=:blue, leg=false)
```

Out[31]:



In [42]:

```
for i=1:20
    z[i]=(fib(i+1)/fib(i))
end
z
```

Out[42]:

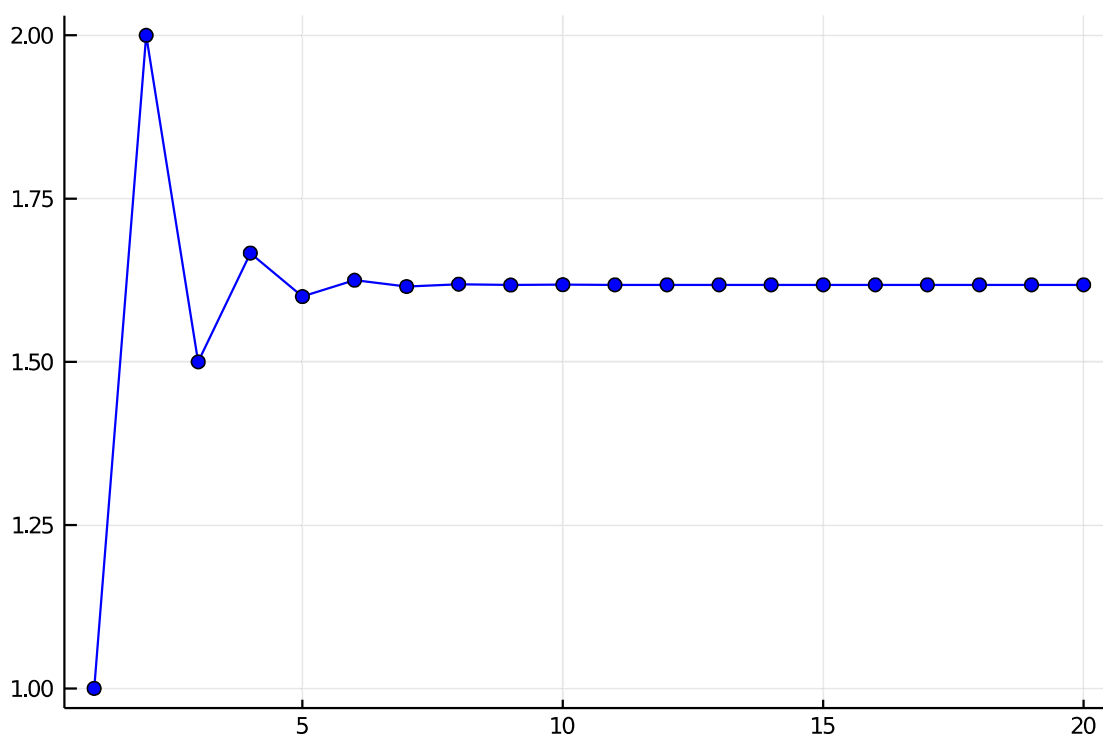
20-element Array{Float64,1}:

```
1.0
2.0
1.5
1.6666666666666667
1.6
1.625
1.6153846153846154
1.619047619047619
1.6176470588235294
1.6181818181818182
1.6179775280898876
1.6180555555555556
1.6180257510729614
1.6180371352785146
1.618032786885246
1.618034447821682
1.6180338134001253
1.618034055727554
1.6180339631667064
1.6180339985218033
```

In [43]:

```
plot(z,marker=:circle, color=:blue, leg=false)
```

Out[43]:



In []: