PSEUDOCODE

Step 1: Declare all necessary values (Basic Reproductive Ratio, grid size, transmission rate, recovery rate) [h=0.25 days]

Step 2: Create arrays for S, I and R.

Step 3: Run a loop to calculate the successive values for S, I and R using the following formulae.

```
s1=(0-b*s[c]*i[c])
i1=(b*s[c]*i[c]-g*i[c])
r1=(g*i[c])
s2=0-b*(s[c]+h*s1/2)*(i[c]+h*i1/2)
i2=b*(s[c]+h*s1/2)*(i[c]+h*i1/2)-g*(i[c]+h*i1/2)
r2=(g*(i[c]+h*i1/2))
s3=0-b*(s[c]+h*s2/2)*(i[c]+h*i2/2)
i3=b*(s[c]+h*s2/2)*(i[c]+h*i2/2)-g*(i[c]+h*i2/2)
r3=(g*(i[c]+h*i2/2))
s4=0-b*(s[c]+h*s3/2)*(i[c]+h*i3/2)
i4=b*(s[c]+h*s3/2)*(i[c]+h*i3/2)-g*(i[c]+h*i3/2)
r4=(g*(i[c]+h*i3/2))
s[c+1] = s[c]+h*(s1+2*s2+2*s3+s4)/6
i[c+1] = i[c]+h*(i1+2*i2+2*i3+i4)/6
r[c+1] = r[c]+h*(r1+2*r2+2*r3+r4)/6
```

Step 4: Increase c for successive values of S, I and R and end the loop when number of infected is zero (below 1).

Step 5: Find maximum value in the array for infected people

Step 6: Find the value corresponding to 1% of the maximum value

Step 7: Find the time duration between the maximum and the 1% value

Step 8: Plot the results as S, I, R vs Time

Code

In [66]:
```
# Atirek Aryan 200221
r=3.65 #Basic Reproductive  Ratio
h=0.25 #grid size
b=1.66 #transmission rate
g=b/r #recovery rate
g
```

Out[66]: 0.45479452054794517

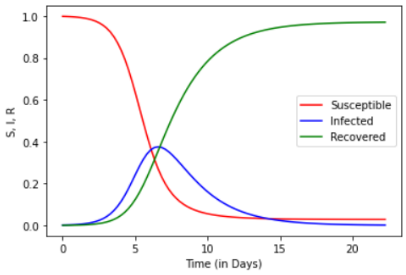
In [67]:
```
# Creating arrays for S, I and R fractional values
s=[]
s.append(762/763)
i=[]
i.append(1/763)
r=[]
r.append(0/763)
```

In [68]:
```
#Running a while loop to compute successive values for the S, I and R

c=0
while(1):
    s1=(0-b*s[c]*i[c])
    i1=(b*s[c]*i[c]-g*i[c])
    r1=(g*i[c])
    s2=0-b*(s[c]+h*s1/2)*(i[c]+h*i1/2)
    i2=b*(s[c]+h*s1/2)*(i[c]+h*i1/2)-g*(i[c]+h*i1/2)
    r2=(g*(i[c]+h*i1/2))
    s3=0-b*(s[c]+h*s2/2)*(i[c]+h*i2/2)
    i3=b*(s[c]+h*s2/2)*(i[c]+h*i2/2)-g*(i[c]+h*i2/2)
    r3=(g*(i[c]+h*i2/2))
    s4=0-b*(s[c]+h*s3/2)*(i[c]+h*i3/2)
    i4=b*(s[c]+h*s3/2)*(i[c]+h*i3/2)-g*(i[c]+h*i3/2)
    r4=(g*(i[c]+h*i3/2))
    s.append(s[c]+h*(s1+2*s2+2*s3+s4)/6)
    i.append(i[c]+h*(i1+2*i2+2*i3+i4)/6)
    r.append(r[c]+h*(r1+2*r2+2*r3+r4)/6)
    c=c+1
    if(i[c]<1/763):
        break
```

In [69]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [70]:
```
#Creating array for time
t=np.linspace(0,22.25,num=89)
```

In [80]:
```
#Plotting S,I and R according to the  data calculated
plt.plot(t, s, 'r', label="Susceptible")
plt.plot(t, i, 'b', label="Infected")
plt.plot(t, r, 'g', label="Recovered")
plt.legend()
plt.xlabel("Time (in Days)")
plt.ylabel("S, I, R")
plt.show()
```

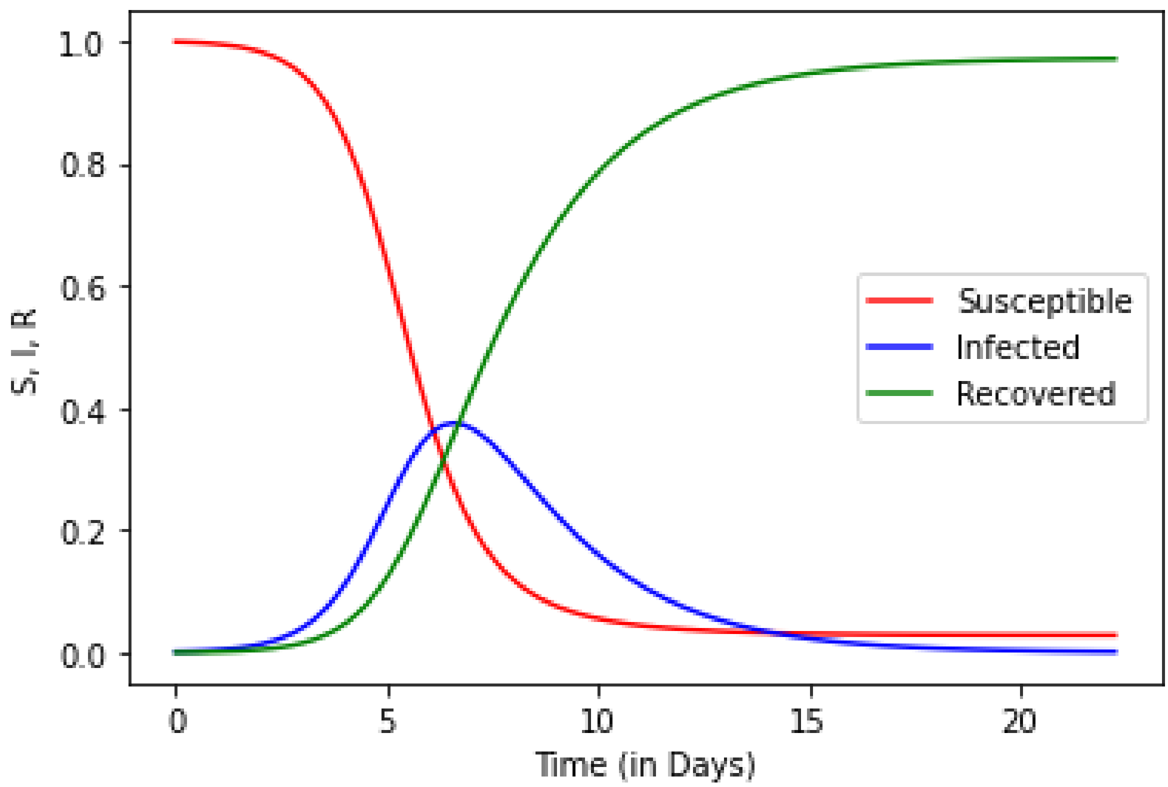


In [72]:
```
#Finding maximum value for Infected Group
max=0
for j in range(89):
    if(i[max]<i[j]):
        max=j
max
```

Out[72]: 26

```
In [73]: i[26]*763
Out[73]: 286.2145844941352

In [74]: #Finding the 1% of the maximum Infected value
         op=0
         for j in range(26,89):
             if(i[26]*0.01>i[j] or i[26]*0.01==i[j] ):
                 op=j
                 break

In [75]: op
Out[75]: 77

In [76]: i[26]
Out[76]: 0.37511741087042616

In [77]: i[77]
Out[77]: 0.003719229085859914

In [78]: i[76]
Out[78]: 0.004122208282372485

In [79]: #Time taken to reach the 1%  from Maximum
         time=(77-26)*0.25
         time
Out[79]: 12.75
```

Graph



Calculations

Max I = 286.21 ~ 287

Time Duration for disease to reach 1% of Maximum = 12.75 days