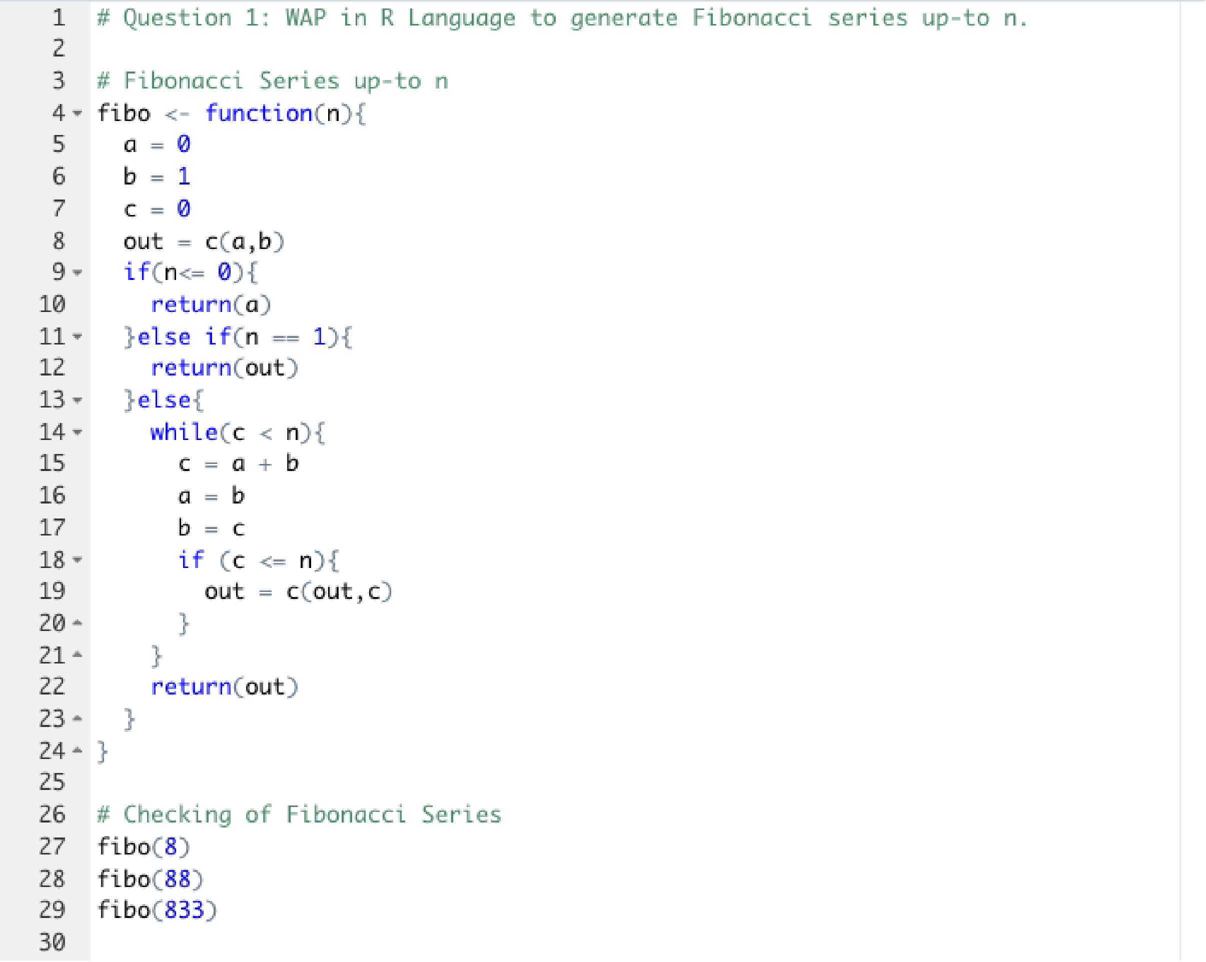
Abhimanyu Bhowmik
A910119819008

Lab 3 Exp 1

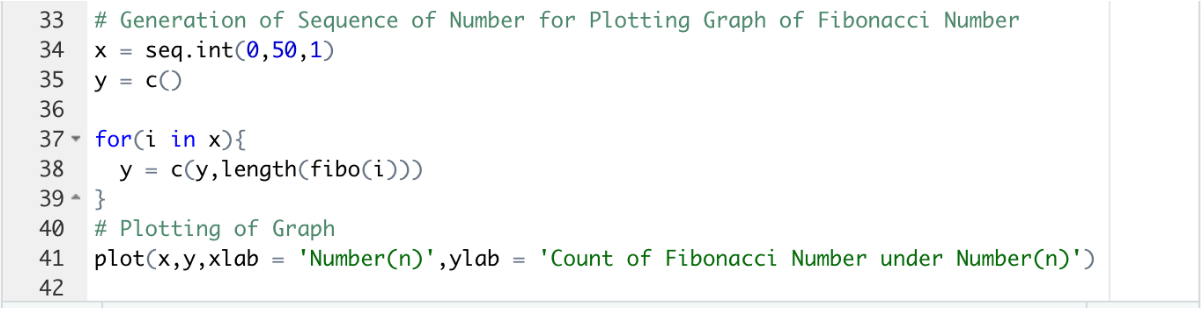
Date: 28.03.2022

AIM: WAP in R Language to generate Fibonacci series upto n.

CODE:

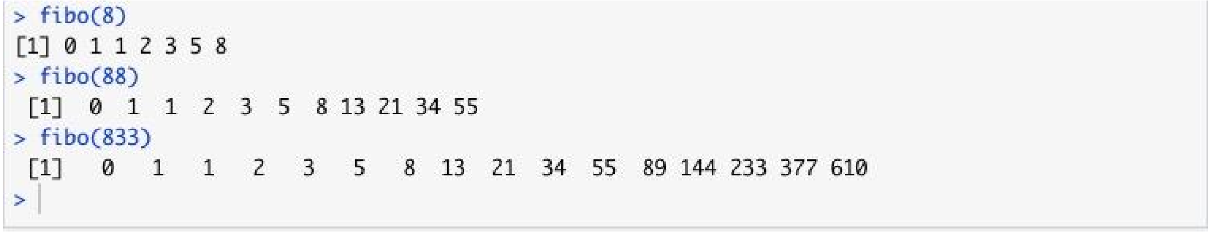


```r
# Question 1: WAP in R Language to generate Fibonacci series up-to n.

# Fibonacci Series up-to n
fibo <- function(n){
  a = 0
  b = 1
  c = 0
  out = c(a,b)
  if(n<= 0){
    return(a)
  }else if(n == 1){
    return(out)
  }else{
    while(c < n){
      c = a + b
      a = b
      b = c
      if (c <= n){
        out = c(out,c)
      }
    }
    return(out)
  }
}

# Checking of Fibonacci Series
fibo(8)
fibo(88)
fibo(833)

```

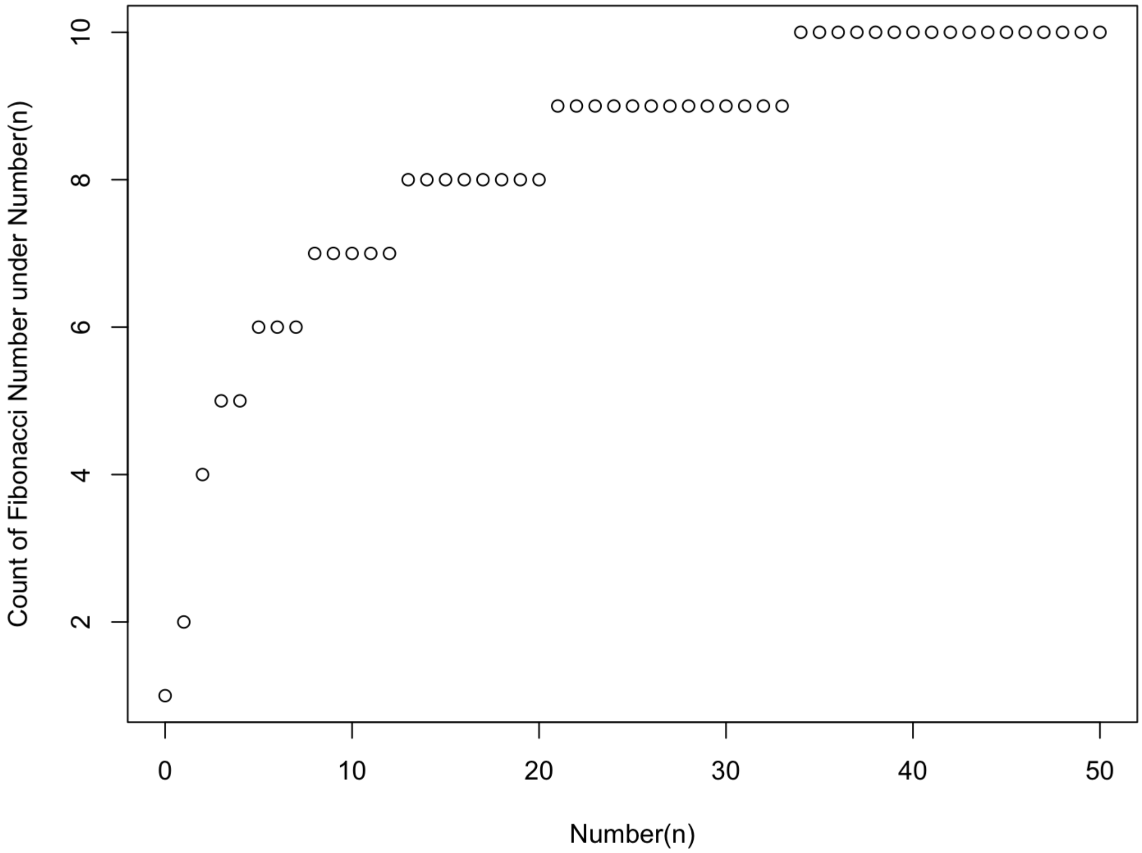


```r
# Generation of Sequence of Number for Plotting Graph of Fibonacci Number
x = seq.int(0,50,1)
y = c()

for(i in x){
  y = c(y,length(fibo(i)))
}
# Plotting of Graph
plot(x,y,xlab = 'Number(n)',ylab = 'Count of Fibonacci Number under Number(n)')

```

OUTPUT:



```
> fibo(8)
[1] 0 1 1 2 3 5 8
> fibo(88)
 [1]  0  1  1  2  3  5  8 13 21 34 55
> fibo(833)
 [1]   0   1   1   2   3   5   8  13  21  34  55  89 144 233 377 610
> 
```

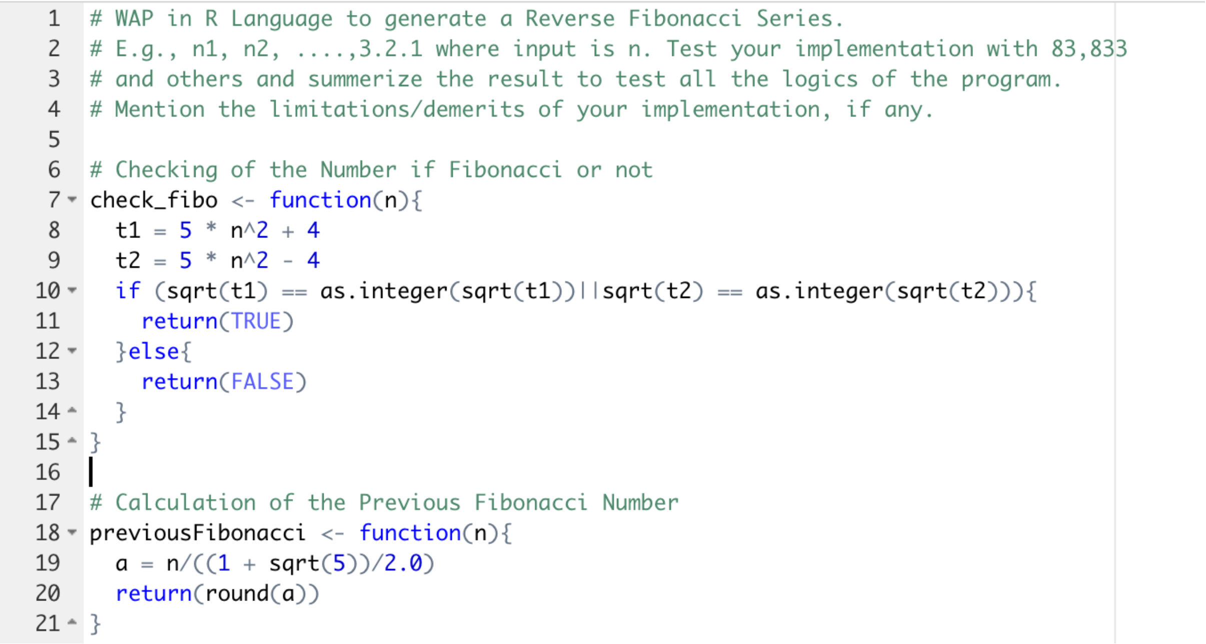


Lab 3 Exp 2

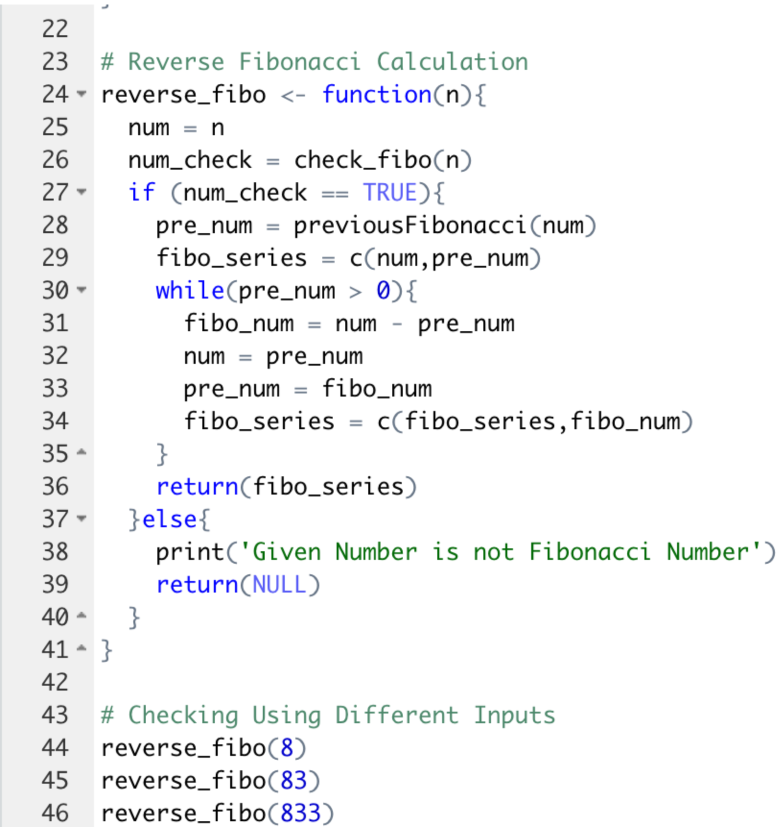
Date: 28.03.2022

AIM: WAP in R Language to generate a Reverse Fibonacci Series. E.g., n1, n2, ...,3.2.1 where input is n. Test your implementation with 8, 83, 833 and others and summerize the result to test all the logics of the program.Mention the limitations/demerits of your implementation, if any.

CODE:

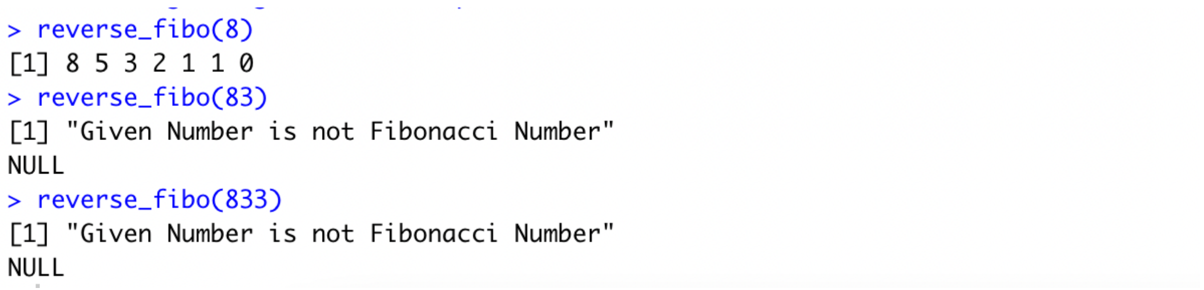


```r
# WAP in R Language to generate a Reverse Fibonacci Series.
# E.g., n1, n2, ....,3.2.1 where input is n. Test your implementation with 83,833
# and others and summerize the result to test all the logics of the program.
# Mention the limitations/demerits of your implementation, if any.

# Checking of the Number if Fibonacci or not
check_fibo <- function(n){
  t1 = 5 * n^2 + 4
  t2 = 5 * n^2 - 4
  if (sqrt(t1) == as.integer(sqrt(t1))||sqrt(t2) == as.integer(sqrt(t2))){
    return(TRUE)
  }else{
    return(FALSE)
  }
}

# Calculation of the Previous Fibonacci Number
previousFibonacci <- function(n){
  a = n/((1 + sqrt(5))/2.0)
  return(round(a))
}
```



```r

# Reverse Fibonacci Calculation
reverse_fibo <- function(n){
  num = n
  num_check = check_fibo(n)
  if (num_check == TRUE){
    pre_num = previousFibonacci(num)
    fibo_series = c(num,pre_num)
    while(pre_num > 0){
      fibo_num = num - pre_num
      num = pre_num
      pre_num = fibo_num
      fibo_series = c(fibo_series,fibo_num)
    }
    return(fibo_series)
  }else{
    print('Given Number is not Fibonacci Number')
    return(NULL)
  }
}

# Checking Using Different Inputs
reverse_fibo(8)
reverse_fibo(83)
reverse_fibo(833)
```

OUTPUT:



```
> reverse_fibo(8)
[1] 8 5 3 2 1 1 0
> reverse_fibo(83)
[1] "Given Number is not Fibonacci Number"
NULL
> reverse_fibo(833)
[1] "Given Number is not Fibonacci Number"
NULL
```