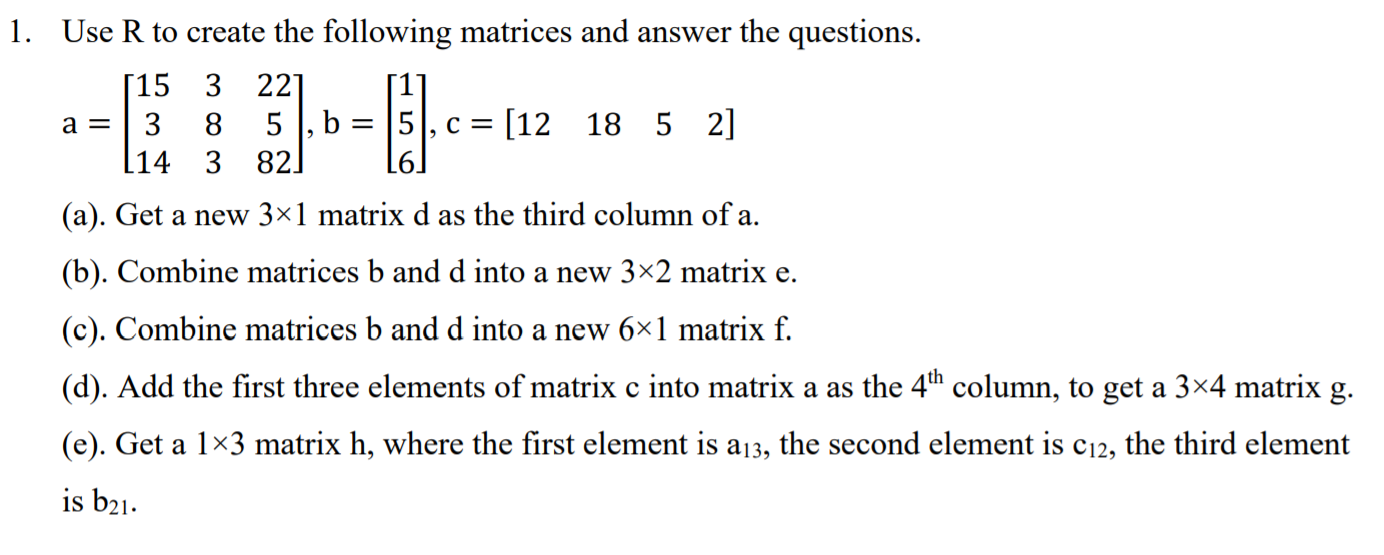
Homework1

Neil Joshi

02/02/2020



CODE:

#Creating the matrices

v1 = c(15,3,14,3,8,3,22,5,82)

v2 = c(1,5,6)

v3 = c(12,18,5,2)

a = matrix(v1,nrow = 3)

b = matrix(v2,nrow = 3)

c = matrix(v3,nrow = 1)

print(a)

print(b)

print(c)

# a)

d = matrix(a[,3])

print(d)

# b)

e = cbind(b,d)

print(e)

# c)

f = rbind(b,d)

print(f)

# d)

g = rbind(a,c[,1:3])

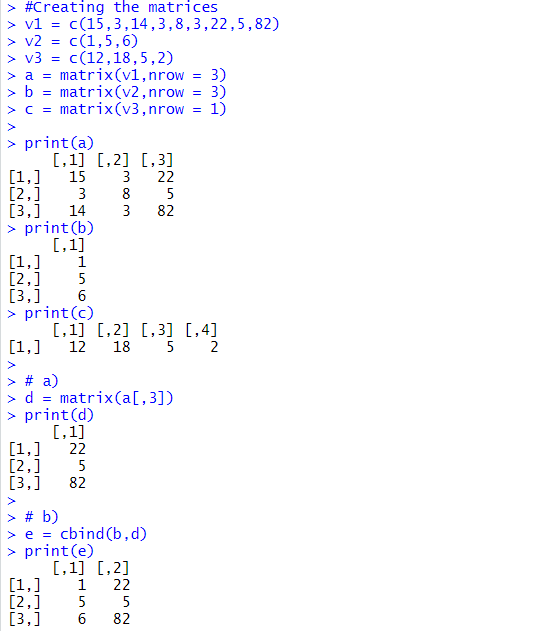
print(g)

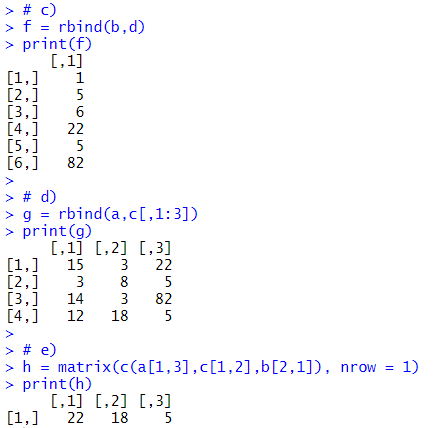
# e)

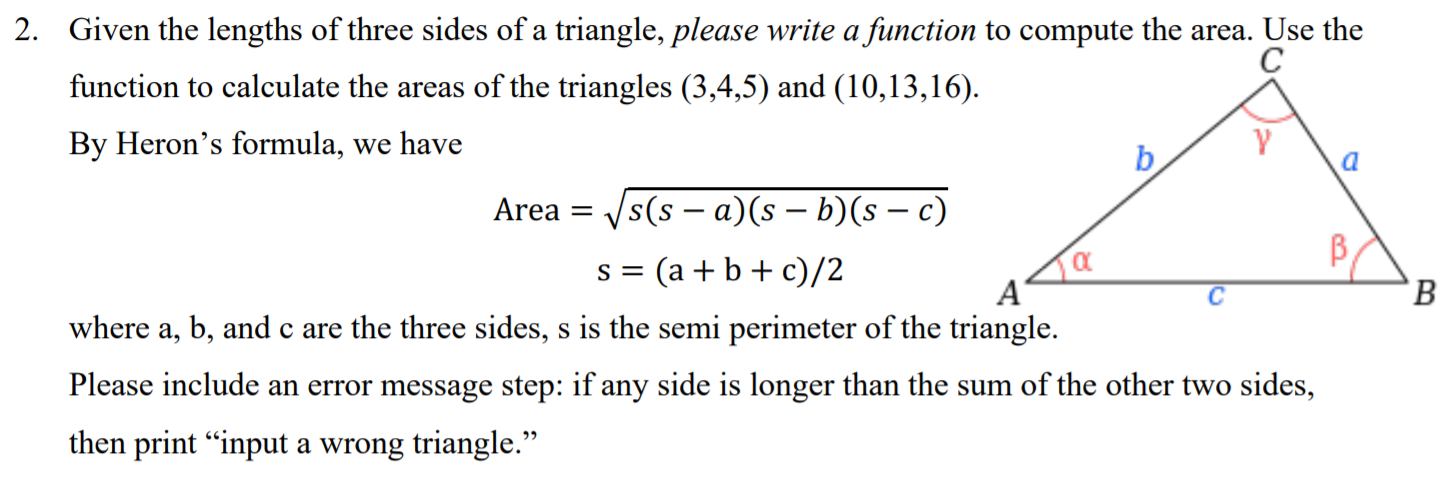
h = matrix(c(a[1,3],c[1,2],b[2,1]), nrow = 1)

print(h)

OUTPUT:







CODE:

Area = function(a,b,c){

#Check longer side condition

# a,b,c are the sides of the triangle

if (max(a,b,c) > (a+b+c-max(a,b,c))){

print('input wrong – invalid triangle')

break

}

#Calculate and print area

s = (a+b+c)/2

area = sqrt(s\*(s-a)\*(s-b)\*(s-c))

print(area)

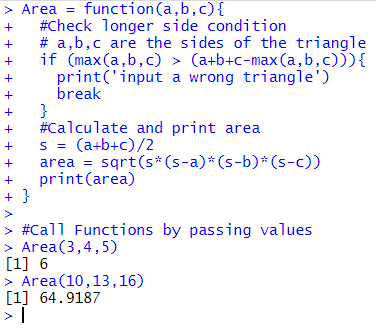
}

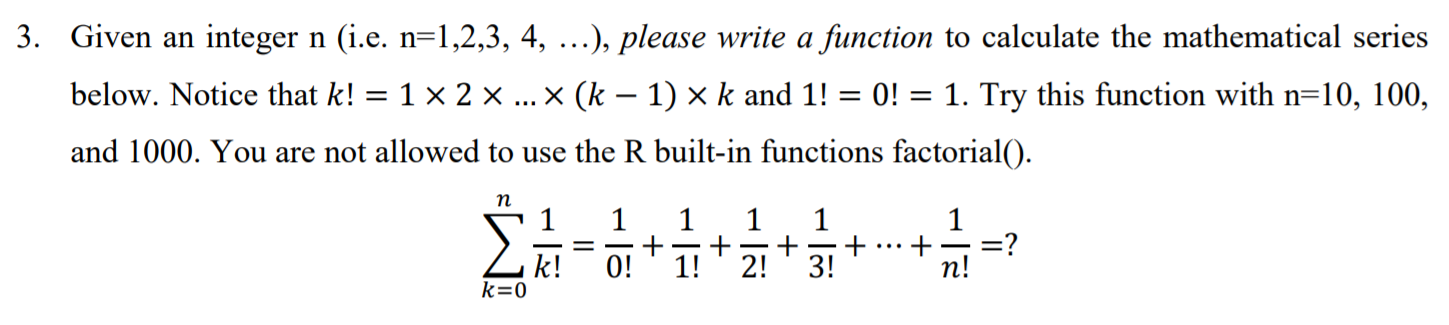
#Call Functions by passing values

Area(3,4,5)

Area(10,13,16)

OUTPUT:





CODE:

#Declare Variable

fact\_num = 1

sum = 0

#Declare a function - Series

Series = function(n){

total\_range = 1:n

for (i in total\_range){

for (j in 1:i){

fact\_num = fact\_num \* j

}

sum = sum + (1/fact\_num)

}

print(sum)

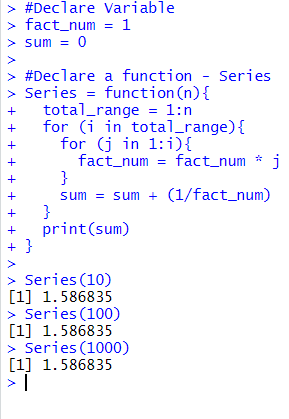
}

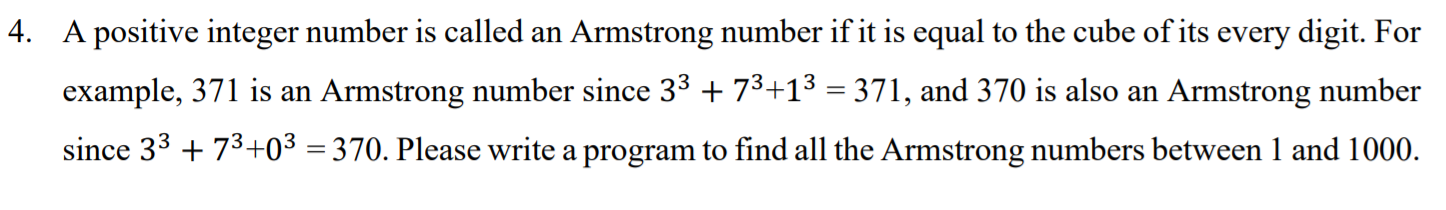
Series(10)

Series(100)

Series(1000)

OUTPUT:





CODE:

for (num in 1:1000) {

sum = 0

temp\_num = num

while (num > 0) {

# Extracting Units digit

q = num %% 10

sum = sum + q \*\* 3

num = num %/% 10

}

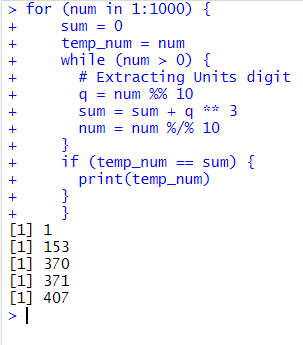
if (temp\_num == sum) {

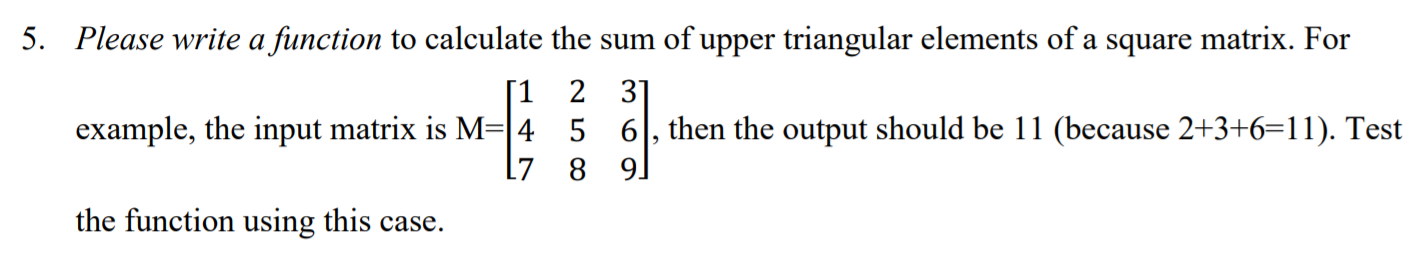
print(temp\_num)

}

}

OUTPUT:





CODE:

Sum\_Upper\_Tri = function(in\_mat){

# Counting no of rowns and col

r = nrow(in\_mat)

c = ncol(in\_mat)

sum = 0

if (r != c) {

print('Not a Square Matrix')

break()

} else {

for (i in 1:r) {

for (j in i:c) {

if (j > i) {

# Code for upper triangle.

sum= sum + in\_mat[i,j]

}

}

}

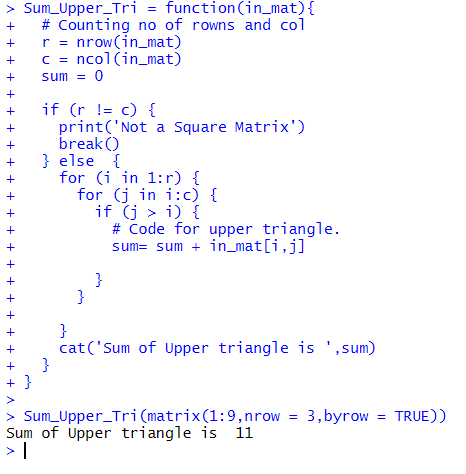
cat('Sum of Upper triangle is ',sum)

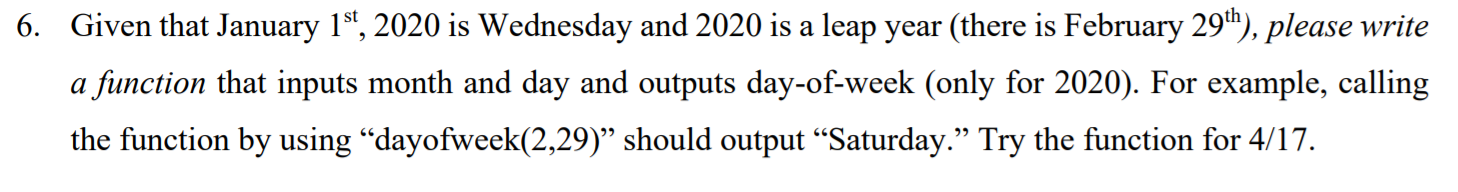
}

}

Sum\_Upper\_Tri(matrix(1:9,nrow = 3,byrow = TRUE))

OUTPUT:





CODE:

dayofweek = function(m,d){

print("Function Valid of year 2020")

total\_days = d

# Declare Vector with total number of days each month

month\_no\_days = c(31,29,31,30,31,30,31,31,30,31,30,31)

# Declare day of week vector w.r.t. 1st Jan being a Wed

day\_num = c('Tue','Wed','Thurs','Fri','Sat','Sun','Mon')

if (m != 1) {

m1 = m - 1

for (i in 1:m1) {

total\_days = total\_days + month\_no\_days[i]

}

}

week\_num = (total\_days) %% 7

print('Day for given date is:')

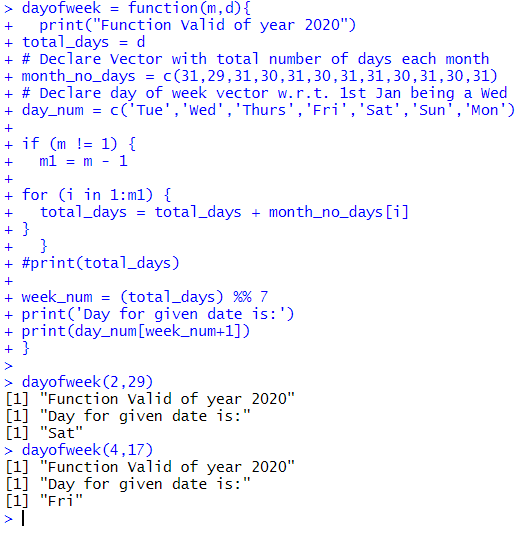
print(day\_num[week\_num+1])

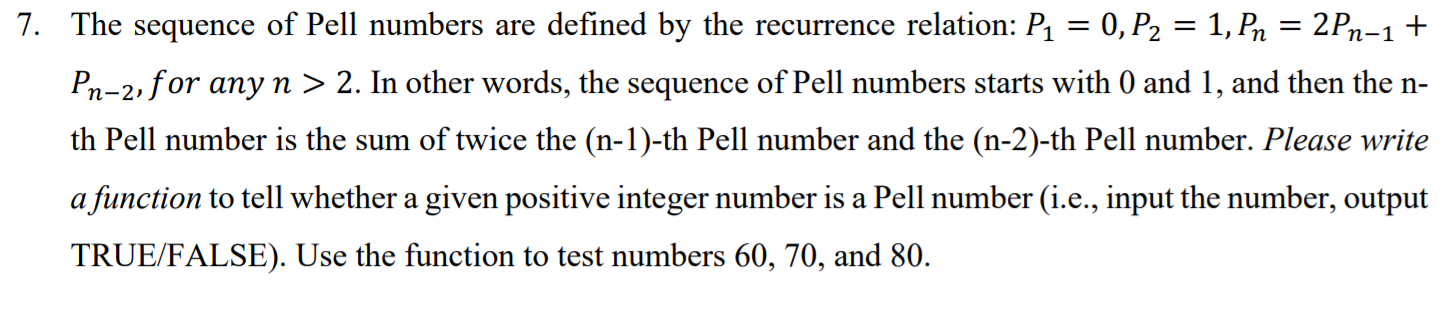
}

dayofweek(2,29)

dayofweek(4,17)

OUTPUT:





CODE:

is\_Pell = function(n){

# Declaring first two given values

p1 = 0

p2 = 1

pn = 0

while (pn < n) {

pn = p1 + 2\*p2

# Interchange p1 and p2 Values

p1 = p2

p2 = pn

#print(pn)

}

if (pn == n) {

print('TRUE’)

} else {

print(‘FALSE')

}

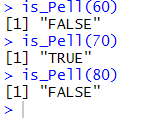
}

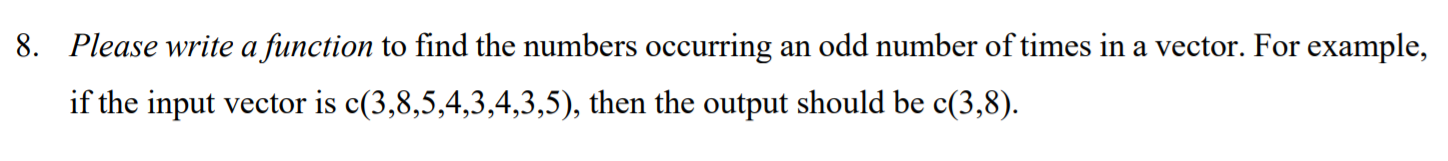
is\_Pell(60)

is\_Pell(70)

is\_Pell(80)

OUTPUT:





CODE:

Odd\_Num = function(v){

new\_vec = vector()

for (i in v) {

count = 0

for (j in v){

if (i == j) {

count = count + 1

}

}

#print(count)

if ((count %% 2 != 0) & !(i %in% new\_vec) ) {

new\_vec = c(new\_vec,i)

}

}

cat('c(',new\_vec,")")

}

Odd\_Num(c(1,2,1,4,2,3,2))

Odd\_Num(c(3,8,5,4,3,4,3,5))

OUTPUT:

