# week2 assignment

# 1. Finding if the number is prime number or not

prime\_number = function(x) {

if(x==1){return("Number is neither prime nor composite")}

else {

if(any(x %% (2:sqrt(x)) == 0)){return("Not a prime number")}

else{return("Prime number")}

}

}

# Method 2 :

prime\_number = function(x) {

if(x==1){return("Number is neither prime nor composite")}

else {

for(i in 2:sqrt(x)) {

if(x %% i == 0){

return("Not a prime number")

break

}

}

return("Prime number")}

}

# 2. Convert a given decimal number to binary

binary = function(x) {

c = ''

decimal = x-as.integer(x)

while(x > 0) {

c = paste0(as.integer(x) %% 2,c)

print(c)

x = as.integer(x/2)

}

d = ''

if(decimal != 0) {

for(i in 1:10) {

decimal = decimal\*2

d = paste0(d,as.integer(decimal))

print(d)

if(as.integer(decimal) == 1){

decimal = decimal-1

}

}

}

return(paste0(c,'.',d))

}

binary(10.2)

# 3. Generate Pascal’s triangle

pascal = function(n) {

for(line in 1:n)

{

c = 1

d = numeric()

for (i in 1:line)

{

d = paste0(unlist(d),c)

c = c \* (line - i) / i

}

print(d)

}

}

pascal(3)

# 4. Print Fibonacci series using loops

fibonacci = function(n) {

a = 0

b = 1

if(n == 0){print(a)}

else {

print(a)

print(b)

for (i in 3:n) {

c = a + b

a = b

b = c

print(c)

}

}

}

fibonacci(5)

# 5. Check whether a number is an Armstrong number or not using a while loop.

# (An Armstrong number, also known as narcissistic number, is a number that is equal to the sum of the cubes of its own digits

armstrong = function(x) {

sum = 0

n = x

while(x > 0) {

temp = x %% 10

sum = sum + (temp^3)

x = as.integer(x/10)

}

if(n == sum){print("Armstrong Number")}

else{print("Not an Armstrong Number")}

}

armstrong(370)