Experiment 4: Conditionals and Functions

### Even Odd Number Test

x <- 20893 # sample input  
if(x %% 2 == 0) {  
 print("Even number detected.")   
} else {   
 print("Odd number detected.")  
}

## [1] "Odd number detected."

print(x)

## [1] 20893

### Prime Composite Number Test

y <- 2293 # sample input  
ctr <- 0  
for(index in seq(1, y)) {  
 if(y %% index == 0) ctr <- ctr + 1  
}  
  
if(ctr == 2) {  
 print("Prime")  
} else {  
 print("Composite")  
}

## [1] "Prime"

### Simple Calculator

print('An expression is of the format x1 op x2, where x1, x2 are two numbers, op is an operator like +, -, \*, /. Please make sure that x1, op, x2 are space seperated.')

## [1] "An expression is of the format x1 op x2, where x1, x2 are two numbers, op is an operator like +, -, \*, /. Please make sure that x1, op, x2 are space seperated."

user\_input <- readline('Specify an expression! ')

## Specify an expression!

parts <- strsplit(user\_input, " ")  
parts <- parts[[1]]  
  
x1 <- as.numeric(parts[1])  
op <- parts[2]  
x2 <- as.numeric(parts[3])  
  
value <- switch (op,  
 "+" = x1 + x2,  
 "-" = x1 - x2,  
 "\*" = x1 \* x2,  
 "/" = x1 / x2  
)  
print(  
 value  
)

## NULL

## TWO THREE SIX

numset <- 1:100  
for(x in numset) {  
 if(x %% 2 == 0 && x %% 3 == 0) print(paste(x, "SIX"))  
 else if(x %% 2 == 0) print(paste(x, "TWO"))  
 else if (x %% 3 == 0) print(paste(x, "THREE"))  
}

## [1] "2 TWO"  
## [1] "3 THREE"  
## [1] "4 TWO"  
## [1] "6 SIX"  
## [1] "8 TWO"  
## [1] "9 THREE"  
## [1] "10 TWO"  
## [1] "12 SIX"  
## [1] "14 TWO"  
## [1] "15 THREE"  
## [1] "16 TWO"  
## [1] "18 SIX"  
## [1] "20 TWO"  
## [1] "21 THREE"  
## [1] "22 TWO"  
## [1] "24 SIX"  
## [1] "26 TWO"  
## [1] "27 THREE"  
## [1] "28 TWO"  
## [1] "30 SIX"  
## [1] "32 TWO"  
## [1] "33 THREE"  
## [1] "34 TWO"  
## [1] "36 SIX"  
## [1] "38 TWO"  
## [1] "39 THREE"  
## [1] "40 TWO"  
## [1] "42 SIX"  
## [1] "44 TWO"  
## [1] "45 THREE"  
## [1] "46 TWO"  
## [1] "48 SIX"  
## [1] "50 TWO"  
## [1] "51 THREE"  
## [1] "52 TWO"  
## [1] "54 SIX"  
## [1] "56 TWO"  
## [1] "57 THREE"  
## [1] "58 TWO"  
## [1] "60 SIX"  
## [1] "62 TWO"  
## [1] "63 THREE"  
## [1] "64 TWO"  
## [1] "66 SIX"  
## [1] "68 TWO"  
## [1] "69 THREE"  
## [1] "70 TWO"  
## [1] "72 SIX"  
## [1] "74 TWO"  
## [1] "75 THREE"  
## [1] "76 TWO"  
## [1] "78 SIX"  
## [1] "80 TWO"  
## [1] "81 THREE"  
## [1] "82 TWO"  
## [1] "84 SIX"  
## [1] "86 TWO"  
## [1] "87 THREE"  
## [1] "88 TWO"  
## [1] "90 SIX"  
## [1] "92 TWO"  
## [1] "93 THREE"  
## [1] "94 TWO"  
## [1] "96 SIX"  
## [1] "98 TWO"  
## [1] "99 THREE"  
## [1] "100 TWO"

### Armstrong Number Test

user\_input <- 153L # sample input  
x\_temp <- user\_input  
dig\_sum <- 0  
while(x\_temp > 0) {  
 digit <- x\_temp %% 10  
 dig\_sum <- dig\_sum + (digit ^ 3)  
 x\_temp <- x\_temp %/% 10  
}  
if(dig\_sum == user\_input) {  
 print("Armstrong number detected.")  
} else {  
 print("Armstrong number not detected.")  
}

## [1] "Armstrong number detected."

## Functions

greet <- function(name) {  
 print(paste('Hello, ', name, '! You are amazing.', sep = ''))  
}  
greet('Dev')

## [1] "Hello, Dev! You are amazing."

### Write a function to get square of number in sequence

x <- 1:20  
squared <- function(i) {  
 return(i\*i)  
}  
squared(x)

## [1] 1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289  
## [18] 324 361 400

### Write a function to print a factors of a number

factors <- function(x) {  
 y <- seq(1, x)  
 return(y[x %% y == 0])  
}  
  
print('Factors of 10 are')

## [1] "Factors of 10 are"

print(factors(10))

## [1] 1 2 5 10

print('Factors of 9190 are')

## [1] "Factors of 9190 are"

print(factors(9190))

## [1] 1 2 5 10 919 1838 4595 9190

### Write a function to count sum of n natural numbers

sumofnatural <- function(n) {  
 return(sum(0:n))  
}  
  
print('Sum of first 10 natural numbers')

## [1] "Sum of first 10 natural numbers"

print(sumofnatural(10))

## [1] 55