##################################################################################

# 변수 삭제하기

var1 = 2

vae2 = 5

rm(list=ls()) # ls() : 환경창에 있는 객체들

cat("\014") # console clear

####################################################################################

student\_midterm=read.table(file.choose(),sep=",",fileEncoding = "UTF-8")

student\_midterm

student\_midterm=read.table(file.choose(),sep=",",fileEncoding = "UTF-8",header = T)

student\_midterm

#####################################################################################

# 파일로부터 데이터를 읽어들일 때 일반 txt형식은 많이 사용되지 않는다

# 일반 txt 형식은 많이 사용되지 않는다

# 컴퓨터간에 데이터를 주고 받으려고 한다

# process간에 데이터 통신을 하기 위해서 특정 형식을 이용해 데이터를 주고 받는다

# 1. CSV( comma seperated value )

# comma 기호를 이용해서 데이터를 구분, 해당 문자열을 전달해서 데이터 통신을 해보자

# 예) "홍길동, 20, 서울, 김길동, 30, 부산, 최길동, 50, 인천,..."

# 장점 : 간단하다, 구조가 간단하여 부가적인 데이터가 적다(상대적으로 크기가 작다)

# -> 많은 양의 데이터를 처리할 수 있다

# 단점 : 구조적 데이터를 표현하기에 적합하지 않다 = parsing 작업이 복잡하다

# 유지보수에 문제가 발생한다

# 2. XML 방식

# tag를 이용해서 데이터를 표현하는 방식

# 예) <name>홍길동</name><age>20</age><address>서울</address>

# <phone>

# <mobile>010-111-2222</mobiel>

# <home>02-333-4444</home>

# </phone>

# 장점 : 구조적 데이터를 표현하기에 적합, 사용하기에 편리, 데이터의 의미를 표현할 수 있다 , tag안에 새로운 tag 추가 가능

# 단점 : 부가적인 데이터가 너어무 크다.

# 3. JSON (Java Script Object Notation)

# 예) {name : "홍길동", age : 20, address : 서울,...}

# 장점 : 구조적 표현이 가능하면서 XML보다 크기가 작다

########################################################################################

# 어제는 read.table() : sep가 있어야한다

# 오늘은 read.csv() : sep가 ","이기 때문에 생략가능, header = T가 기본

getwd()

df = read.csv(file.choose(),fileEncoding = "UTF-8")

df

# Excel 파일을 불러올 수 있다.

# 확장 pacakge를 이용해야 한다.

# R을 설치하면 base system이 설치된다고 표현한다( base package, recommended package )

# other package

# xlslx package를 설치하고 로딩한다

install.packages("xlsx")

library(xlsx)

# JAVA\_HOME 환경변수 설정

Sys.setenv(JAVA\_HOME="C:\\Program Files\\Java\\jre1.8.0\_231")

student\_midterm = read.xlsx(file.choose(),sheetIndex = 1, encoding = "UTF-8")

student\_midterm

class(summary(student\_midterm))

#######################################################################################

# 처리된 결과를 file에 write 해보자!

# write.table() : data frame을 file에 저장

# cat() : 분석 결과 (vector)를 file에 저장

# capture.output() : 분석 결과(list,table)를 file에 저장

cat("처리된 결과는 :","\n","\n", file = "C:/R\_R/Data/report.txt", append = T)

#\n은 한 칸 띄운다는 의미

#append=T는 원래 파일에 덧붙여서 추가한다는 의미! F면 새로 덮어씌워서 기존 데이터가 날라감

write.table(student\_midterm,file= "C:/R\_R/Data/report.txt",row.names = F, quote = F, append = T)

capture.output(summary(student\_midterm),file = "C:/R\_R/Data/report.txt", append = T )

#row.names=F : 행 번호 샂게, quote : "" 삭제

#write.xlsx()

df = data.frame(x=c(1:5),y=seq(1,10,2), z=c("a","b","c","d","e"),stringsAsFactors=F)

df

#stringsAsFactors = F 데이터가 범주형이 아닌 문자형

write.xlsx(df, "C:/R\_R/Data/report.xlsx")

#if구문(조건문)

# 조건에 따라서 실행되는 code block을 제어할 수 있다.

var1 = 100

var2 = 20

if(var1>var2) {

#조건문이 참일 때 실행

cat("참이에요!")

} else {

#조건문이 거짓일 때 실행

cat("거짓이에요!")

}

#ifelse : Java의 3항 연산자와 같다. 조건 ? 맞으면 1 틀리면 2번째

var1 = 10

var2 = 20

ifelse(var1>var2,"참일경우 선택","거짓일 경우 선택")

# 반복문(for,while)

# for : 반복 횟수만큼 반복 실행

# while : 조건이 TRUE일 동안 반복 실행

for(var1 in seq(1:5)){

print(var1)

}

idx = 1

mySum = 0

while(idx<10){

mySum = mySum + idx

idx = idx+1

}

mySum

#45

##############로직으로 문제풀기 (제어문 이용 : if,ifelse, for, while)#############3

# 1. 1부터 100 사이에 있는 3의 배수를 출력하자

for( var1 in seq(1:100)){

if( var1 %% 3 == 0){

print(var1)

}

}

# 2. 1부터 100 사이에 있는 prime number()를 출력하자

for(var1 in seq(1:100)){

if( is.integer(var1%/%2)=T)

}

#for( var1 in seq(1:100)){

# if(var1%/%var2

#######################################################################################

# 사용자 정의 함수 (User Define Function)

# 함수명 = function(x) {...}

# 입력받은 숫자를 제곱해서 돌려주는 함수를 하나 만들어보자

myFunc = function(x){

x = x\*x

return(x)

}

var1 = myFunc(5)

var1 = c(1:10)

sum(var1)

# sum 함수와 동일한 역할을 하는 mySum을 만들어보자

# vector를 입력으로 받아서 합을 구하는 함수

mySum = function(x){

result = 0

for(t in x){

result = result + t

}

return(result)

}

var1=c(1:10)

mySum(var1)

#######################3

mySum = function(x)

result = 0

for(var1 in c(1:10)){

result = result + x

}

mySum(var2)

for(var1 in seq(1:100)){

for(var2 in seq(2:50)){

var3 = var1/var2

if(is.integer(var3)=T & var3=1){

print(var1)

}

}

}

####################################################################################

#연습문제 풀기

# 사용할 데이터 : 2 3 5 6 7 10

# 1. 주어진 데이터로 vector x를 생성하세요

x <- c(2,3,5,6,7,10); x

# 2. 주어진 데이터 각각을 제곱해서 vector x를 생성하세요

v = c(2,3,5,6,7,10)

x = v\*v; x

x = v^2

#결과 : 4 9 25 36 49 100

# 3. 주어진 데이터 각각을 제곱해서 합을 구하세요.

x = c(2,3,5,6,7,10)

sum(x^2)

#결과 : 223

# 4.주어진 데이터에서 5보다 큰 값들로 구성된 vector x를 구하세요

v = c(2,3,5,6,7,10)

x = v[v>5]

x

x = c(2,3,5,6,7,10)

a = c(2,3,5,6,7,10) > 5 #mask #fancy indexing

x[a]

#6 7 10

# 5. vector x의 길이를 구하세요

length(x)

#6

# 6.R의 package중에 UsingR 이 있어요!

install.packages("UsingR")

library(UsingR)

# 1부터 2003까지 숫자 중 prime number를 가지고 있다.

data("primes")

head(primes) #앞에서부터 6개만 출력

tail(primes) #뒤에서부터 6개만 출력

# 7. 1부터 2003까지 숫자 중 prime number 는 몇개인가요?

length(primes)

# 8. 1부터 200까지 숫자 중 prime number는 몇개인가요?

sum(primes < 201)

#결과 : 46

# 9. 평균을 구해보세요

mean(primes)

#결과 : 917.9375

# 10. 500 이상 1000이하의 prime number 만으로 구성된 vector p를 구하세요

x = c(primes)

p = c(primes >= 500 & primes <= 1000)

x[p]

#결과 :

#[1] 503 509 521 523 541 547 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 673

#[28] 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 823 827 829 839 853 857 859

#[55] 863 877 881 883 887 907 911 919 929 937 941 947 953 967 971 977 983 991 997

# 11. 다음과 같은 형태의 데이터를 이용하여 아래의 문제를 풀어보아요...!

# 1 5 9

# 2 6 10

# 3 7 11

# 4 8 12

# 11-1. matrix x를 생성하세요

x = matrix(c(1:12),ncol=3); x

#12. 전치행렬을 만들어보아요

t(x)

# 13. matrix x에 대해 첫번째 행만 추출하세요

x[1,]

# 14. matrix x에 대해 6,7,10,11을 matrix 형태로 추출하세요

x[2:3,2:3]

# 15. matrix x에 대해 x의 두번째 열의 원소가 홀수인 행들만 뽑아서 matrix p를 생성하세요

p = x[x[,2] %% 2 == 1, ]; p

################################프로그래밍 로직로직로직로직##################################

#문법, 자료구조 활용해보자 띵.ㅋ...ㅡ...

# 홀수개의 숫자로 구성된 숫자문자열이 입력으로 제공됩니다.

# 문자열의 개수는 7개 이상 11개 이하로 제한됩니다.

# -> 입력문자열의 길이는 7 or 9 or 11

# 중앙 숫자를 기준으로 앞과 뒤의 숫자를 분리한 후 분리된 두 수를 거꾸로 뒤집어서 두 수의 차를 구한다

#

# 예) 7648623 -> 764 623(가운데 수는 버림) -> 467 326 -> 467-326 = 141

# 변수 두개, 인덱스로 추출 시 c사용, 거꾸로 도출?

# 예) 764862 -> 764 620(가운데 수는 버림) -> 467 026 -> 467-26 = 441

# 문자를 숫자로 전환 #문자열을 벡터 7개로 나눠줄 수 없나

library(stringr)

input = "7648623"

n\_input = as.numeric(str\_split(input,""))

l\_input = str\_length(input)

n\_input

#길이가 조건에 맞으면 인덱스값으로 데이터 추출해서 값을 구한다 아니면 안함

if(l\_input %% 2 == 1 & l\_input >= 7 & l\_input <=9 ){

first = l\_input[c(3:1)]

second = l\_input[c(6:4)]

result = first-second

print(result)

}

#########################################################################################

# web

# 데이터를 구축

# Database, 파일로부터 데이터를 얻어올 수 있다.

# 프로그램적으로 web scraping, crawling을 통해 데이터를 구축

# HTML, CSS, JavaScript

# HTML : 웹페이지의 내용, 구조를 담당

# CSS : 웹페이지의 styling을 담당

# JavaScript : 웹페이지의 동적처리를 담당

# 기본적으로 web은 CS구조로 이루어져있다.

# CS구조 : client-server 구조

# server쪽 프로그램 : web server, HTML,CSS,JavaScript를 제공

# client쪽 프로그램 : chrome(browser)

# 서버로부터 HTML, CSS,JavaScript를 받아서 화면에 rendering을 한다.

# HTML, CSS, JavaScript를 이용해서 클라이언트에게 제공해주는 web page를 만들어보자

# 개발툴이 있어야한다.

# eclipse 가 가장 일반적이지만 webstorm을 이용해보자

# 클라이언트에게 제공할 HTML을 작성했다(소리없는 아우성)

# 서버프로그램(web server)에게 우리 프로젝트를 알려줘야한다!

# 우리 프로젝트를 web server에게 인지시켜야한다. (configure시킨다)

# web server program을 기동시켜서 우리 프로젝트를 웹에 deploy

# web client(browser)를 이용해서 URL(주소, 좌표)서버쪽 프로그램에 접속해서 HTML을 받아가서 화면에 그리자

# web server가 webstorm에 내장되어있다!