다음에 제시된 문제들을 R 코드로 작성한 후에 lab\_07.R 로 저장하여 제출합니다.

# 문제1

~~~~

# 문제2

~~~~

[ 문제1 ]

다음 사양의 함수 countEvenOdd() 을 생성한다.

매개변수 : 1 개

리턴값 : 리스트

기능 : 숫자벡터를 아규먼트로 받아 짝수의 갯수와 홀수의 갯수를 카운팅하여

리스트(각 변수명 : even, odd)로 리턴한다.

전달된 데이터가 숫자 백터가 아니면 NULL 을 리턴한다.

[ 문제2 ]

다음 사양의 함수 vmSum() 을 생성한다.

매개변수 : 1 개

리턴 값 : 숫자벡터

기능 : 전달받은 아규먼트가 벡터인 경우에만 기능을 수행한다.

벡터가 아니면 “벡터만 전달하숑!”라는 메시지를 리턴한다.

벡터라 하더라도 숫자 벡터가 아니면 “숫자 벡터를 전달하숑!” 라는

메시지를 출력하고 0 을 리턴한다.

전달된 숫자 벡터의 모든 값을 더하여 리턴한다.

[ 문제3 ]

다음 사양의 함수 vmSum2() 을 생성한다.

매개변수 : 1 개

리턴 값 : 숫자벡터

기능 : 전달받은 아규먼트가 벡터인 경우에만 기능을 수행한다.

벡터가 아니면 “벡터만 전달하숑!”라는 메시지를 가지고 error 를 발생시킨다.

벡터라 하더라도 숫자 벡터가 아니면 “숫자 벡터를 전달하숑!” 라는

메시지를 가지고 warning 을 발생시키고 0 을 리턴한다.

전달된 숫자 벡터의 모든 값을 더하여 리턴한다.

[ 문제4 ]

다음의 기능을 지원하는 함수 mySum()을 생성한다.

아규먼트 : 벡터 한 개

리턴값 : 리스트 한 개 또는 NULL

(1) 전달된 벡터에서 짝수번째 데이터들의 합과 홀수번째 데이터들의 합을 구하여

list 객체로 리턴하는데 각각 oddSum과 evenSum 이라고 변수명을 설정한다.

(2) 벡터가 온 경우에만 기능을 수행하며 벡터가 오지 않은 경우에는 "벡터만 처리 가능!!"이라는

메시지로 에러를 발생시킨다.

(3) 전달된 벡터에 NA 값이 하나라도 존재하는 경우에는 "NA를 최저값으로 변경하여 처리함!!" 이라는

메시지를 경고를 발생시킨다. 그리고 NA 는 최저값으로 설정하여 기능을 수행한 후에 결과를 리턴한다.

(4) NULL이 온 경우에는 NULL을 리턴한다.

[ 문제5 ]

다음의 기능을 지원하는 함수 myExpr()을 생성한다.

아규먼트 : 함수 한 개

리턴값 : 한 개의 숫자값

(1) 아규먼트로 함수를 전달받는다.

(2) 아규먼트가 함수가 하니면 "수행 안할꺼임!!" 이라는 메시지로 에러를 발생시킨다.

(3) 1부터 45 사이의 난수 6개를 추출하여 아규먼트로 전달된 함수를 호출하고 그 결과를

리턴한다.

[ 문제6 ]

다음 사양의 함수 createVector1() 을 생성한다.

아규먼트 : 가변(숫자, 문자열, 논리형(데이터 타입의 제한이 없다.))

리턴 값 : 벡터

(1) 전달된 아규먼트가 없으면 NULL을 리턴한다.

(2) 전달된 아규먼트에 하나라도 NA 가 있으면 NA를 리턴한다.

(3) 전달된 데이터들을 가지고 벡터를 생성하여 리턴한다.

[ 문제7 ]

다음 사양의 함수 createVector2() 을 생성한다.

매개변수 : 가변(숫자, 문자열, 논리형(데이터 타입의 제한이 없다.))

리턴 값 : 리스트객체

기능 : 전달된 아규먼트가 없으면 NULL을 리턴한다.

전달된 데이터들을 각 타입에 알맞게 **각각의** 벡터들을 만들고 리스트에 담아서 리턴한다.

[ 문제8 ] – 함수 문제 아님

iotest1.txt 파일에 저장된 데이터들을 읽고 다음과 같은 형식으로 결과를 출력하는

R 코드를 구현하고 test1.R 로 저장하여 제출한다.

오름차순 : ….

내림차순 : ….

합 : ...

평균 : ...

[ 문제9 ] – 함수 문제 아님

iotest2.txt 파일에 저장된 데이터들을 읽고 다음과 같은 형식으로 결과를 출력하는

R 코드를 구현하고 test2.R 로 저장하여 제출한다.

“가장 많이 등장한 단어는 XX 입니다.”