## using a pair of function that store and on=btain the value

## of cached inverse matrix or evaluates the inverse if cache not exists.

## The first function makeCacheMatrix creates a special "matrix"

## which is actually a list of functions to set and get value of the matrix

## and also to set and get the value of inverse of the matrix

makeCacheMatrix <- function(x = matrix()) {

makeCacheMatrix <- function(x = matrix()) {

m <- NULL

set <- function(y){

x <<- y

m<<-NULL

}

get <- function() x

setmat <- function(solve) m<<-solve

getmat <- function() m

list(set=set, get=get, setmat=setmat, getmat=getmat)

}

## The second function, below either retrieves cached inverse of the special "matrix"

## or calculates the inverse value and prints it

cacheSolve <- function(x, ...) {

## Return a matrix that is the inverse of 'x'

m <- x$getmat()

if(!is.null(m)) {

message ("Getting the cached data")

return (m)

}

data <- x$get()

m <- solve(data, ...)

x$setmat(m)

m

}