```
#Q1
dmat <- matrix(scan(),nrow=3,byrow=TRUE)
208 24 44 379 100 66 156 85 24

dimnames(dmat) <- list(c("Like it","Don't Care","Don't like it"),
  c("Fans","On reserve","Off reserve"))

percents(dmat,2)

counts <- as.vector(dmat)
opinion <- factor(row(dmat),labels=dimnames(dmat)[[1]])
group <- factor(col(dmat),labels=dimnames(dmat)[[2]])

indepFit <- glm( counts ~ opinion + group , family=poisson)

stdResid <- matrix(rstandard(indepFit),nrow=3,dimnames=dimnames(dmat))
stdResid

ordFit <-  glm( counts ~ group + opinion + group:(as.integer(opinion)),
  family=poisson())
satFit <- glm( counts ~ opinion * group , family=poisson)

anova(indepFit,ordFit,satFit, test="Chi")


fitMat <- matrix(fitted(ordFit),nrow=3,dimnames=dimnames(dmat))
round(percents(fitMat,2,pretty=FALSE),1)

#Q2
dmat <- matrix(scan(),nrow=3,byrow=TRUE)
5 4 1 12 42 14 2 14 10

types <- c("Arches","Loops","Worls")
dimnames(dmat) <- list(types,types)

chisq.test(dmat)
fisher.test(dmat)

percents(apply(dmat,1,sum))
percents(apply(dmat,2,sum))

library(concord)
stuart.maxwell.mh(dmat)


#Q3

m <- matrix(scan(),nrow=2,byrow=TRUE)
17
2
5
26

prop.test(x=apply(m,1,sum)[1],n=sum(m),correct=FALSE)

prop.test(x=apply(m,2,sum)[1],n=sum(m),correct=FALSE)

mcnemar.test(m)

depend.prop.ci <- function(X, succ.index=1, alpha=.05){
# Function from Simonoff's web page.
# Produces confidence interval for the difference between dependent proportions.
# Data are input as a 2 x 2 matrix; by default, the success category corresponds
# first row and column
#
# Default level is 95% confidence
#
  i <- succ.index
  z <- qnorm(1 - alpha/2)
  n <- sum(X)
  propmatrix <- X/n
  v <- sqrt((sum(propmatrix[i,])*(1 - sum(propmatrix[i,])) + sum(propmatrix[,i])*
    2.*(propmatrix[1,1]*propmatrix[2,2] - propmatrix[1,2]*propmatrix[2,1]))/n)
  est <- sum(propmatrix[i,]) - sum(propmatrix[,i])
  low <- est - z*v
  high <- est + z*v
  list(est=est, low=low, high=high)}

unlist(depend.prop.ci(m))


  row.props <- as.matrix(apply(m, 1, sum)/sum(m))
  col.props <- as.matrix(apply(m, 2, sum)/sum(m))
  cell.props <- m/sum(m)
  expected <- t(row.props) %*% col.props
   observed <- sum(diag(cell.props))
  kappaStat <- (observed - expected )/(1-expected)

freqArray <- array(scan(), dim=c(2,2,2),
  dimnames=list(cad=c("yes","no"),ht=c("yes","no"),age=c("<49",">65")))
550 212
941 495
1102 87
1018 109

freqArray <- aperm(freqArray,c(2,1,3))

apply(freqArray,3,
 function(mat) (mat[1,1]*mat[2,2])/(mat[1,2]*mat[2,1]))
apply(freqArray,3,
 function(mat) (mat[1,1]/(mat[1,1]+mat[1,2]))/(mat[2,1]/(mat[2,1]+mat[2,2])))

mantelhaen.test(freqArray)
```