```
###### Exercice 1 ##########
## Question 1 ###    01 pt
df=read.table("Data.csv", sep=";", header=TRUE)
## Question 2 ###   01 pt
apply(df,2,class)
## Question 3 ###   01 pt
df1=(df[df$Region=="Centre",])
rownames(df1)[which.max(df1$Distance)]
## Question 4 ###  01 pt
df=df[,c(2,1,3:ncol(df))]
## Question 5 ###  01 pt
df[order(df$Region,-df$Population),]  01 pt
## Question 6 ###  02 pts
v=levels(df$Region)
v[v=="Ouest"]="Int"; v[v=="East"]="Ouest"
v[v=="Int"]="East"; levels(df$Region)=v
## Question 6 ### 01 pt
write.table(df,"Data2.csv",sep=";")

###### Exercice 2 ##########
## Question 1 ###  02 pts
tri.inf=function(A) {
for(i in 1:nrow(A)) for(j in 1:ncol(A))
if(i>j && A[i,j]!=0) return(FALSE)
return(TRUE)
}
tri.sup=function(A) {
for(i in 1:nrow(A)) for(j in 1:ncol(A))
if(i<j && A[i,j]!=0) return(FALSE)
return(TRUE)
}
triang=function(A) tri.inf(A)||tri.sup(A)

## Question 2####   02 pts
outerB=function(x,y,f) {
M=matrix(0,nrow=length(x),ncol=length(y))
for(i in 1:nrow(M)) for(j in 1:ncol(M)) M[i,j]=f(x[i],y[j])
return(M)
}


## Question 3 ####  02 pts
nb=function(x) {
n=length(x)
s=0
for(i in 1:(n-1)) if(x[i]*x[i+1]<0) s=s+1
return(s)
}
## Question 4 ####   01 pt
nb2=function(x) sum(x[1:(length(x)-1)]*x[2:length(x)]<0)

## Exercice 3 ###
### Vecteur 1###   01.5   pts
1-cumsum(1:9)
### Vecteur 2###    01.5 pts
cumsum(1:9)*(2*(1:0)-1)
cumsum(1:9)*c(1,-1)
### Vecteur 3###  01.5 pts
```

```
rep(99:0,rep(1:2,50))
### Matrice 1###  01.5 pts
matrix(8:1,nrow=6,ncol=4,byrow=TRUE)



### Matrice 2###  01 pt
A=matrix(1:4,nrow=6,ncol=6)
M=t(apply(M,1,rev))
diag(M)=0
(M=t(apply(M,1,rev)))
### Deuxième methode
M=matrix(1:4,ncol=6,nrow=6)
X=cumsum(c(6,rep(5,5)))
(M[X]=0)

### Matrice 3###  01 pt
matrix(rep(c("A","B"),1:2),4,5)
###Exercice 4###
## Question 1 ##  02 pts
is.prime=function(k)  {
if(k<2) return(FALSE)
if(k==2) return(TRUE)
for(i in 2:(k/2)) if(k%%i==0) return(FALSE)
return(TRUE)
}
## Question 2 ##  02 pts
primes=function(n){
tab=1:n
cond=sapply(tab,is.prime)
return(tab[cond])
}
Primes=primes(100)
## Question 3 ##   02 pts
#### Conjecture #####
#### Avec boucle ####
N=1000
conj=TRUE
for(k in seq(2,N,by=1))
{
test=FALSE
for(i in 1:length(Primes)) for(j in 1:length(Primes))
if(k==Primes[i]+Primes[j]) {cond=TRUE;break}
if(!cond) {conj=FALSE;break}
}
## Question 4 ##  02 pts
### Sans boucle ####
M=10000
Primes=primes(M)
vec=as.numeric(outer(Primes,Primes,"+"))
vec=unique(sort(vec))
vec=vec[vec%%2==0 & vec<=M]
vec=vec%/%2
L=max(vec)
sum(abs(vec-2:L))==0
```