#EJERCICIO 1.

```
N=100
n=3
SF=c(1,1)
while(n<=N){
 SF[n]=SF[n-2]+SF[n-1]
 n=n+1
}
SF
```

#EJERCICIO 2.

```
N=10
M=10
n=1
MAT=matrix(c(0),nrow = M,ncol=N)
V=SF
while(n<=N){
 MAT[1,n]=n
 MAT[2,n]=V[n]
 MAT[3,n]=MAT[1,n]*n
  for(i in 4:M){
    MAT[i,n]=(MAT[i-1,n])^(n)
  }
 n=n+1
}
MAT
```

#EJERCICIO 3

```
x=matrix(c(0), nrow = n, ncol=4)
y=matrix(c(0), nrow = n, ncol=4)
n=10
h=4
a=2
```

```
b=1
C=0
r=matrix(c(0), nrow = n, ncol=4)
g=matrix(c(0), nrow = n, ncol=4)
X=matrix(c(0), nrow = 4*n+3*4, ncol=1)
Y=matrix(c(0), nrow = 4*n+3*4, ncol=1)
#el numero de filas de X,Y está pensado para que haya
#huecos de filas para ver mejor el resultado
for (i in 1:n){
  g[i,1]=(i-1)*h
  g[i,2]=(3*pi)/4+(i-1)*h
  g[i,3]=pi+(i-1)*h
  g[i,4]=(7*pi)/4+(i-1)*h
}
for (j in 1:4){
#Para poder distinguir las coordenadas de cada cuadrante:
  t=c("Empieza el cuadrante número")
  t2=c("1","2","3","4")
#para dejar un hueco entre cuadrantes
  X[b,1]=""
  Y[b,1]=""
  b=b+1
  X[b,1]=t
  Y[b,1]=t2[j]
  b=b+1
  X[b,1]=""
  Y[b,1]=""
  b=b+1
  for (i in 1:n){
   r[i,j]=a*sqrt(abs(2*cos(2*g[i,j])))
```

```
#he puesto el valor absoluto porque si no da error NaN
  x[i,j]=r[i,j]*cos(g[i,j])
  y[i,j]=r[i,j]*sin(g[i,j])
  X[b,1]=x[i,j]
  Y[b,1]=y[i,j]
  b=b+1
 }
}
C=data.frame(X,Y)
C
#EJERCICIO 4
M=7
n=8
x=6
E=0
fact=1
S=0
P=1
F=c(1,3,6,4,6,2,3,1)
s=c(3,5,4,4,3,2,3,5)
for(n in 1:M){
 for(i in 1:n){
   P=P*(x-s[i])
 }
 fact=fact*n
 S=S+(1/(fact))*(F[n]*P)
}
E=S
E
```

#EJERCICIO 5

```
N=0
s=0
A=5
M=65
V=c(0)
while(s<=M){
  N=N+1
  s=s+N
  V[N]=N
}
U=matrix(c(0),nrow=(N-1), ncol=4)
#nrow=(N-1) porque no va a coger el último valor N
#que es el que hace superar a M
for(i in 1:(N-1)){
  U[i,1]=V[i]
  U[i,2]=V[i]
  U[i,3]=U[i,1]+U[i,2]
  U[i,4]= A*(U[i,1]+U[i,2]+U[i,3])
}
U
```