

# Índice

---

EJERCICIO 1 .....2

EJERCICIO 2 .....6

EJERCICIO 3 .....7

## EJERCICIO 1

- Script

```
#EJERCICIO 1
#=====

Saber si un número es par
#-----
EsPar<- function(m)
{
  {
    if (m%%2==0) {sol<-"el número es par"}
    else {sol<- "el numero es impar"}
  }
  sol
}

v<- c(7, 3, 2, 6, 5, 9)
#.....
EsPar2<- function(v)
{
  contador<- 0
  for (i in (1:length(v))) {
    if (v[i]%%2==0) {contador<- contador+1}
  }
  return (contador)
}
EsPar2(v)

"

# Calcular el factorial de 8
Factorial<- function(n)
{
  f<- 1
  while (n>0)
  {
    f<- f*n
    n<- n-1
  }
  return(f)
}
#-----
#Construir un vector añadiendo elementos por la derecha
Miagenda<- function()
{
  continuar<- TRUE
  while (continuar==TRUE)
  {
    name.in<- readline(prompt = "Introducir el nombre ")
    numero.in<- readline(prompt = "Introducir el número de teléfono ")
    continuar<- readline(prompt = "¿Desea añadir más nombres y tlf?(TRUE/FALSE) ")
  }
  v<- data.frame(name.in,numero.in)
  return(v)
}
Miagenda()
agenda<- Miagenda()
#-----
```

```
####Construir una agenda de teléfonos#####

nueva.agenda<-function(){
##esta función crea una agenda que almacenará en un dataframe con dos campos: nombres, telefonos
nombres<-"Victoria"
telefonos<-"234432234"
continuar<-"S";
while (continuar == "S") {
  nombre<-readline(prompt="Dime el nombre: ")
  telefono<-readline(prompt="Dime el teléfono: ")
  nombres<-c(nombres,nombre)
  telefonos<-c(telefonos,telefono)
  continuar <- readline(prompt="¿Continuar (S/N)? ")
}
mdf<-data.frame(nombres, telefonos)
mdf$nombres<-as.character(mdf$nombres)
mdf$telefonos<-as.character(mdf$telefonos)
return (mdf)
}
miagenda<-nueva.agenda()

####Construir una agenda de teléfonos#####

add.agenda<-function(agenda.ant){
##esta función añade nombres y teléfonos a una agenda que está almacenada en un dataframe con dos campos: nombres, telefonos
nombres<-agenda.ant$nombres
telefonos<-agenda.ant$telefonos
continuar<-"S";
while (continuar == "S") {
  nombre<-readline(prompt="Dime el nombre: ")
  telefono<-readline(prompt="Dime el teléfono: ")
  nombres<-c(nombres,nombre)
  telefonos<-c(telefonos,telefono)
  continuar <- readline(prompt="¿Continuar (S/N)? ")
}
return (data.frame(nombres, telefonos))
}

miagenda2<-add.agenda(miagenda)

#-----

#-----
#Construir un vector que contenga los n números pares siguientes a m (par)
SiguientesPares<-function(n, m)
{
  if (m%%2==0){
    #considero que el numero m introducido es par, en caso contrario el vector sera NA
    v<- vector()
    j<- 2
    for (i in (1:n)){
      numero<- (m+j)
      v[i]<- numero
      j<- j+2}
    }
    return(v)
  }
}
SiguientesPares(5, 30)
```

- Resultados de la consola

```
#-----
EsPar<- function(m)
{
  {
    if (m%%2==0) {sol<-"el número es par"}
    else {sol<- "el numero es impar"}
  }
  sol
}
EsPar(5)
l] "el numero es impar"
|
```

```

> v<- c(7, 3, 2, 6, 5, 9)
> #.....
> EsPar2<- function(v)
+ {
+   contador<- 0
+   for (i in (1:length(v))) {
+     if (v[i]%%2==0) (contador<- contador+1)}
+   return (contador)
+ }
> EsPar2(v)
[1] 2
> |

```

```

> #-----
> # Calcular el factorial de 8
> Factorial<- function(n)
+ {
+   f<- 1
+   while (n>0)
+   {
+     f<- f*n
+     n<- n-1
+   }
+   return(f)
+ }
> factorial(9)
[1] 362880
> |

```

```

> nueva.agenda<-function(){
+   ###esta función crea una agenda que almacenará en un dataframe con dos campos$
+   nombres<-"Victoria"
+   telefonos<-"234432234"
+   continuar<-"S";
+   while (continuar == "S") {
+     nombre<-readline(prompt="Dime el nombre: ")
+     telefono<-readline(prompt="Dime el teléfono: ")
+     nombres<-c(nombres,nombre)
+     telefonos<-c(telefonos,telefono)
+     continuar <- readline(prompt="¿Continuar (S/N)? ")
+   }
+   mdf<-data.frame(nombres, telefonos)
+   mdf$nombres<-as.character(mdf$nombres)
+   mdf$telefonos<-as.character(mdf$telefonos)
+   return (mdf)
+ }
> miagenda<-nueva.agenda()
Dime el nombre: Ana
Dime el teléfono: 111
¿Continuar (S/N)? S
Dime el nombre: Adrian
Dime el teléfono: 222
¿Continuar (S/N)? N
> |

```

```
miagenda
  nombres telefonos
Victoria 234432234
      Ana      111
      Adrian   222
```

---

```
> add.agenda<-function(agenda.ant){
+ ###esta función añade nombres y teléfonos a una agenda que está almacenada en un
+ nombres<-agenda.ant$nombres
+ telefonos<-agenda.ant$telefonos
+ continuar<-"S";
+ while (continuar == "S") {
+ nombre<-readline(prompt="Dime el nombre: ")
+ telefono<-readline(prompt="Dime el teléfono: ")
+ nombres<-c(nombres,nombre)
+ telefonos<-c(telefonos,telefono)
+ continuar <- readline(prompt="¿Continuar (S/N)? ")
+ }
+ return (data.frame(nombres, telefonos))
+ }
```

```
>
> miagenda2<-add.agenda(miagenda)
Dime el nombre: Maria
Dime el teléfono: 333
¿Continuar (S/N)? N
> miagenda2
  nombres telefonos
1 Victoria 234432234
2      Ana      111
3  Adrian   222
4   Maria    333
```

```
> #-----
> #Construir un vector que contenga los n números pares siguientes a m (par)
> SiguientesPares<-function(n, m)
+ {
+ if (m%%2==0){
+ #considero que el numero m introducido es par, en caso contrario el vector sera NA
+ v<- vector()
+ j<- 2
+ for (i in (1:n)){
+ numero<- (m+j)
+ v[i]<- numero
+ j<- j+2}
+ }
+ return(v)
+ }
> SiguientesPares(5, 30)
[1] 32 34 36 38 40
>
```

## EJERCICIO 2

```
#Ejercicio 2
#=====

#Realizar las modificaciones pertinentes en el último ejemplo para que también funcione si el
# número m es impar, almacenar todo en un nuevo script y ejecutarlo para ver su funcionamiento. Para ello usar
# 'Interpretar código fuente' del menú de consola o ejecutar el comando: source("ruta/fichero.R")

SiguientesPares3<-function(n, m)
{
  #considero que el numero m introducido es impar, en caso contrario el vector sera NA
  v<- vector()
  {
    if (m%%2!=0) {j<- 1}
    else {j<- 2}
  }
  for (i in (1:n)){
    numero<- (m+j)
    v[i]<- numero
    j<- j+2}
  return(v)
}

> SiguientesPares3<-function(n, m)
+ {
+ #considero que el numero m introducido es impar, en caso contrario el vector sera NA
+ v<- vector()
+ {
+ if (m%%2!=0) {j<- 1}
+ else {j<- 2}
+ }
+ for (i in (1:n)){
+ numero<- (m+j)
+ v[i]<- numero
+ j<- j+2}
+ return(v)
+ }
> SiguientesPares3(5, 31)
[1] 32 34 36 38 40
> SiguientesPares3(5, 30)
[1] 32 34 36 38 40
.
|
```

## EJERCICIO 3

- Script

```
#Ejercicio 3
#=====
#Programar funciones para:
# a) Calcular el máximo de un vector sin utilizar 'max'
# b) Calcular la frecuencia de un número x en un vector v de números
# c) Calcular la frecuencia del máximo en un vector.

#apartado a
#-----
# calcular el máximo de un vector
MaxVector<- function(v)
{
  maximo<- v[1]
  for (i in (2:length(v))) {
    if (maximo< v[i]) (maximo = v[i])
  }
  return (maximo)
}

#apartado b
#-----
FrecuenciaNumero<- function(v, x)
{
  frecuencia<- 0
  for (i in (1:length(v)))
  {
    if (v[i] == x) (frecuencia = frecuencia + 1)
  }
  return (frecuencia)
}

#apartado c
#-----
FrecuenciaDelMaximo<- function(v)
{
  numeromaximo<- MaxVector(v)
  return(FrecuenciaNumero(v, numeromaximo))
}
```

- Consola

```
> #Programar funciones para:
> # a) Calcular el máximo de un vector sin utilizar 'max'
> # b) Calcular la frecuencia de un número x en un vector v de números
> # c) Calcular la frecuencia del máximo en un vector.
>
> #apartado a
> #-----
> # calcular el máximo de un vector
> MaxVector<- function(v)
+ {
+ maximo<- v[1]
+ for (i in (2:length(v))) {
+ if (maximo< v[i]) (maximo = v[i])
+ return (maximo)
+ }
> v<- c(1, 2, 3, 5, 8, 2)
> MaxVector(v)
[1] 8
```

```

> #apartado b
> #-----
>
> FrecuenciaNumero<- function(v, x)
+ {
+ frecuencia<- 0
+ for (i in (1:length(v)))
+ {
+ if (v[i] == x) (frecuencia = frecuencia + 1)
+ }
+ return (frecuencia)
+ }
> FrecuenciaNumero(v, 2)
[1] 2
~ |

> #apartado c
> #-----
>
> FrecuenciaDelMaximo<- function(v)
+ {
+ numeromaximo<- MaxVector(v)
+ return(FrecuenciaNumero(v, numeromaximo))
+ }
> FrecuenciaDelMaximo(v)
[1] 1
> |

```