

Tarea 3: Estructuras de datos

Jhan Kevin

23/4/2020

Ejercicio 1

Cread un vector llamado “Harry” formado por la sucesión de números consecutivos entre el -10 y 27. Pedidle a R que os devuelva el elemento de índice 7. Escribid el resultado.

```
Harry <- -10:27
Harry
```

```
## [1] -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8
## [20] 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
```

El valor del vector Harry en la posición 7 es -4

Ejercicio 2

Dad el máximo de la sucesión $(100 \cdot 2^n - 7 \cdot 3^n)$ con $n = 0, \dots, 200$

```
n <- 0:200

fun_s <- function(x){
  100*2^x-7*3^x
}

max <- max(sapply(n, FUN = fun_s))
```

El valor máximo de la sucesión es 1499

Ejercicio 3

Cread la sucesión de números consecutivos entre 0 y 40. A continuación, cread el vector $(3 \cdot 5^n - 1)$ con $n = 0, \dots, 40$. Ponedle como nombre x. Ahora, dad el subvector de los elementos que son estrictamente mayores que 3.5

```
n = 0:40
x <- 3*5^n-1
x[x>3.5]
```

```
## [1] 1.400000e+01 7.400000e+01 3.740000e+02 1.874000e+03 9.374000e+03
## [6] 4.687400e+04 2.343740e+05 1.171874e+06 5.859374e+06 2.929687e+07
## [11] 1.464844e+08 7.324219e+08 3.662109e+09 1.831055e+10 9.155273e+10
## [16] 4.577637e+11 2.28818e+12 1.144409e+13 5.722046e+13 2.861023e+14
## [21] 1.430511e+15 7.152557e+15 3.576279e+16 1.788139e+17 8.940697e+17
## [26] 4.470348e+18 2.235174e+19 1.117587e+20 5.587935e+20 2.793968e+21
## [31] 1.396984e+22 6.984919e+22 3.492460e+23 1.746230e+24 8.731149e+24
## [36] 4.365575e+25 2.182787e+26 1.091394e+27 5.456968e+27 2.728484e+28
```

Ejercicio 4

Cread una función que os devuelva la parte real, la imaginaria, el módulo, el argumento y el conjugado de un número, mostrando solo 2 cifras significativas

RECOMENDACIÓN: En algún momento os hará falta utilizar vectores

```
f <- function(x){  
  print(c(Re(x),Im(x),Mod(x),Arg(x),Conj(x)),2))  
}  
f(5)  
  
## [1] 5 0 5 0 5
```

Ejercicio 5

Cread una función que resuelva ecuaciones de segundo grado (de la forma $Ax^2 + Bx + C = 0$). No importa, por ahora, que tengáis en cuenta las que no tienen solución

RECOMENDACIÓN: En algún momento os hará falta utilizar vectores

```
f = function(A,B,C){  
  c((-B+sqrt(B^2-4*A*C))/(2*A),(-B-sqrt(B^2-4*A*C))/(2*A))  
}  
  
f(2,1,0)  
  
## [1] -0.75 -1.25
```

Ejercicio 6

Tomando el vector $\text{vec} = c(0,9,98,2,6,7,5,19,88,20,16,0)$, dad 3 opciones diferentes para calcular el subvector $c(9,19,20,16)$

```
vec = c(0,9,98,2,6,7,5,19,88,20,16,0)  
  
svec1 <- vec[vec==9 | vec==19 | vec==20 | vec==16]  
svec1  
  
## [1] 9 19 20 16  
  
svec2 <- vec[vec >= 9 & vec <= 20]  
svec2  
  
## [1] 9 19 20 16  
  
svec3 <- vec[c(2,8,10,11)]  
svec3  
  
## [1] 9 19 20 16
```

Tomando el vector vec definido en el apartado anterior, buscad

- qué entradas son pares
- qué entradas no son pares y mayores que 20
- dónde toma vec su valor máximo
- dónde toma vec sus valores mínimos