

```
1 # Pregunta 1
2 print('Pregunta 1')
3 ### Crear un vector con la sucesión de números del -10 al 27
4 Harry <- seq(-10, 27) # nolint: object_name_linter.
5 ### Le pedimos la posición 7 del vector
6 cat("El valor del vector en la posición 7 es: ", Harry[7], "\n")
7
8 # Pregunta 2
9 print('Pregunta 2')
10 ### Definir la función para calcular el valor de la sucesión
11 funcion_sucesion <- function(n) {
12   100 * 2^n - 7 * 3^n
13 }
14
15 ### Calcular la sucesión para n = 0, ..., 200
16 valores_sucesion <- sapply(0:200, funcion_sucesion)
17
18 ### Encontrar el índice del máximo valor
19 indice_maximo <- which.max(valores_sucesion)
20
21 ### Mostrar el valor máximo
22 valor_maximo <- valores_sucesion[indice_maximo]
23
24 ### Mostrar el resultado
25 cat("El máximo de la sucesión se alcanza en n = ", indice_maximo, "\n")
26 cat("El valor máximo es: ", valor_maximo, "\n")
27
28 # Pregunta 3
29 print('Pregunta 3')
30 ### Crear un vector con la sucesión de números del 0 al 40
31 numeros <- seq(0, 40)
32
33 ### Definir la función para calcular el valor de la sucesión
34 funcion_sucesion <- function(n) {
35   3 * 5^n - 1
36 }
37
38 ### Calcular la sucesión para n = 0, ..., 40
39 x <- sapply(numeros, funcion_sucesion)
40 ### Seleccionar los elementos de x que son mayores a 3.5
41 x_mayores_3_5 <- x[x > 3.5]
42 ### Mostrar el subvector
43 print(x_mayores_3_5)
44
45 # Pregunta 4
46 print('Pregunta 4')
47 numero_complejo <- function(z) {
48   ## Extraer parte real e imaginaria
49   re <- Re(z)
50   im <- Im(z)
51
52   ## Calcular módulo y argumento
53   modulo <- round(Mod(z), 2)
54   argumento <- round(Arg(z), 2)
55
56   ## Calcular conjugado
57   conjugado <- complex(real = re, imaginary = -im)
58
59   ## Crear vector con los resultados
60   resultado <- c(re, im, modulo, argumento, conjugado)
61   names(resultado) <- c("Parte real", "Parte imaginaria", "Módulo", "Argumento", "Conjugado")
62
63   return(resultado)
64 }
65
66 print(numero_complejo(3+4i))
67
68 # Pregunta 5
69 print('Pregunta 5')
70
71 resolver_ecuacion_2grado <- function(A, B, C) {
72   ## Calculamos el discriminante
73   discriminante <- B^2 - 4 * A * C
74
75   ## Verificamos si tiene soluciones reales
76   if (discriminante < 0) {
```

```

77 mensaje <- "La ecuación no tiene soluciones reales"
78 soluciones <- NA
79 } else {
80 ### Calculamos las soluciones
81 x1 <- (-B + sqrt(discriminante)) / (2 * A)
82 x2 <- (-B - sqrt(discriminante)) / (2 * A)
83 soluciones <- c(x1, x2)
84 mensaje <- "Las soluciones son:"
85 }
86
87 ### Creamos un vector con los resultados
88 resultado <- c(mensaje, soluciones)
89 names(resultado) <- c("Mensaje", "Solución 1", "Solución 2")
90
91 return(resultado)
92 }
93
94 print(resolver_ecuacion_2grado(1, -5, 6))
95
96 # Pregunta 6
97 print('Pregunta 6')
98
99 vec = c(0,9,98,2,6,7,5,19,88,20,16,0)
100
101 ### Opcion 1
102 vec[c(2, 8, 10, 11)]
103 ### Opcion 2
104 vec[vec %in% c(9, 19, 20, 16)]
105 ### Opcion 3
106 subset(vec, vec %in% c(9, 19, 20, 16))
107
108 ### Identificar entradas pares, entradas no pares y mayores que 20, valor máximo y valores mínimos de
109 vec <- c(0, 9, 98, 2, 6, 7, 5, 19, 88, 20, 16, 0)
110 ### Entradas pares
111 pares <- vec[vec %% 2 == 0]
112 print(pares)
113 ### Entradas no pares y mayores que 20
114 no_pares_mayores_20 <- vec[vec %% 2 != 0 & vec > 20]
115 print(no_pares_mayores_20)
116 ### Posición del valor máximo
117 print(which.max(vec))
118 ### Posiciones de los valores mínimos
119 print(which(vec == min(vec)))
120
121 # Pregunta 7
122 print('Pregunta 7')
123
124 A <- matrix(c(1, 3, 2, 4), nrow = 2, ncol = 2)
125 A_transpuesta <- t(A)
126 A_suma <- A + A_transpuesta
127 A_cuadrado <- A %*% A
128 print(A_suma %*% A)
129
130 # Pregunta 8
131 print('Pregunta 8')
132 B <- matrix(c(2, 0, -6, 0, 6, 5, 0, -2, 5), nrow = 3, ncol = 3, byrow = TRUE)
133 resultado <- eigen(B)
134 print(resultado$values)
135
136 # Pregunta 9
137 print('Pregunta 9')
138 C <- matrix(c(-48, 35, -12, -134, 95, -32, -194, 133, -44), nrow = 3, ncol = 3, byrow = TRUE)
139 resultado <- eigen(C)
140 vectores_propios <- round(resultado$vectors, 3)
141 print(vectores_propios)
142
143 # Pregunta 10
144 print('Pregunta 10')
145 D <- matrix(c(-2, -8, -2, 3, -3, -6, -1, 2, -5, -7, -3, 7, -18, -44, -8, 15), nrow = 4, ncol = 4, byrow = TRUE)
146 rango_D <- rank(D)
147 print(rango_D)

```

