

## BUCLES

### Solución:

1) Lo primero que tenemos que hacer es introducir los datos, es decir, definimos los vectores  $v$  y  $w$ .

A continuación, almacenamos en una variable 'n' la longitud del vector  $v$  (que es igual a la del vector  $w$ ). De esta manera, a la hora de hacer el bucle, en lugar de poner un número concreto como valor final ponemos 'n' consiguiendo así una programación más general que valga para vectores de diferentes dimensiones.

**Error frecuente:** un fallo muy tonto pero muy común es, a la hora de definir  $n$ , en lugar de escribir `length` escribimos `lenght`. Por ello debemos tener especial cuidado en este paso y revisarlo antes de ejecutar, ya que, al haber una diferencia tan pequeña entre las dos palabras, cuando nos da error nos cuesta darnos cuenta de que el fallo está ahí.

Lo siguiente que debemos hacer es inicializar a 0 el vector  $S$  (que es la suma que queremos obtener).

Ahora debemos escribir el bucle.  $i$  será el valor que irá cambiando y este debe variar entre 1 (el primer elemento de cada vector) y 'n' (el último elemento de cada vector). En la siguiente línea escribimos el proceso, la operación que debe realizarse. En este caso queremos obtener un vector  $S$  con la suma de los componentes de  $v$  y  $w$ ; es decir, el primer componente de  $S$  será la suma de los primeros componentes de  $v$  y  $w$ , el segundo la suma de los segundos, y así sucesivamente.

Por último, escribimos `S` o `print(S)` para que el programa nos lo de como dato de salida.



```
> #Ejemplo
> #1
>
> v<-c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> S=0
>
> for(i in 1:n){
+   S[i]=v[i]+w[i]
+ }
> S
[1] 24.00000 -2.75000 82.00000 26.08906
>
> #Comprobamos
> v+w
[1] 24.00000 -2.75000 82.00000 26.08906
>
```

2) Lo primero que tenemos que hacer es, al igual que en el apartado anterior, inicializar la suma C a 0.

Posteriormente comenzamos el bucle. i varía de nuevo desde 1 hasta n. En este caso, en el proceso tenemos que poner a ambos lados de la igualdad la misma variable, ya que cada elemento del vector se va sumando a todos los anteriores.

Y por último escribimos print(C) para que escriba el valor como dato de salida.

```
> #2
>
> C=0
> for(i in 1:n){
+   C=C+v[i]
+ }
> print(C)
[1] 32.7
>
```

3) El producto escalar se obtiene sumando el producto de los componentes en primera posición de ambos vectores, más el producto de los componentes en segunda posición, más el producto de los componentes en tercera posición... y así sucesivamente. Por tanto, en el proceso del bucle también debemos escribir la misma variable a ambos lados de la igualdad, ya que el producto de dos elementos se va sumando a todos los anteriores.

No debemos olvidar inicializar p (como hemos llamado al producto escalar) a 0.

```
> #3
>
> p=0
> for(i in 1:n){
+   p=p+v[i]*w[i]
+ }
> print(p)
[1] 666.4253
>
> #Comprobamos
> v%*%w
      [,1]
[1,] 666.4253
>
```

4) Al multiplicar los vectores componente a componente el resultado será otro vector al que nosotros hemos llamado e. Recordamos que lo primero que se debe hacer es inicializarlo a 0.

Iniciamos el bucle en el que i varía desde 1 hasta n y el proceso sería prácticamente igual a la suma del apartado 1). El primer componente del vector e será el producto de los primeros componentes de los vectores v y w, el segundo componente de e el producto de los segundos componentes...y así hasta el producto de los componentes en posición n.

```

> #4
>
> e=0
> for(i in 1:n){
+   e[i]=v[i]*w[i]
+ }
> print(e)
[1] 144.0000 -0.7500 385.0000 138.1753
>
> #Comprobamos
> v*w
[1] 144.0000 -0.7500 385.0000 138.1753
>

```

5) En este apartado el propio enunciado nos define la operación que debe hacerse en el bucle y los valores entre los que varía j. Por lo que solo habría que escribirlo en el script con la estructura adecuada de bucles y sin olvidar inicializar z a 0 antes de empezar el bucle.

```

> #5
>
> z=0
> for (j in 1:n){
+   z[j]=v[j]+2*w[j]
+ }
> print(z)
[1] 36.00000 -2.50000 159.00000 33.47811
>

```

6) Para construir tablas se emplea la instrucción data.frame. Esta permite almacenar datos de diferentes tipos.

Lo primero que debemos hacer es construir vectores con los datos que queremos que aparezcan en nuestra tabla. En nuestro caso la primera columna debe contener los nombres 'Suma' y 'Producto escalar', por lo que creamos un vector, al que hemos llamado nombres, formado por estos elementos. La segunda columna contiene los resultados de los apartados 2 y 3, los vectores C y p; por lo que creamos otro vector, llamado valores, formado por los resultados C y p.

Y por último, construimos la tabla mediante la instrucción data.frame, en la que escribimos los vectores que forman dicha tabla.

```

> #6
>
> nombres<-c('suma', 'producto escalar')
> valores<-c(C,p)
> Tabla<-data.frame(nombres, valores)
> Tabla
      nombres  valores
1         suma  32.7000
2 producto escalar 666.4253
>

```