

```
1: .data
2:     array: .space 20  # Reservar 20 bytes (5 palabras de 4 bytes cada una) para el arr
ay
3:     size: .word 5    # Tamaño del array
4:     space: .asciiz " "      # Cadena que representa un espacio
5:
6: .text
7: .globl main
8: main:
9:     # Inicializar registros para llenar el array
10:    la    $t0, array      # Cargar la dirección base del array en $t0
11:    lw    $t1, size       # Cargar el tamaño del array en $t1
12:    li    $t2, 1          # Inicializar el valor a 1
13:    li    $t3, 0          # Inicializar índice del array a 0
14:
15: fill_array:
16:    # Condición de salida del bucle
17:    beq   $t3, $t1, end_fill # Si $t3 == $t1, salir del bucle
18:
19:    # Almacenar el valor en el array
20:    sw    $t2, 0($t0)      # Guardar el valor de $t2 en la posición array[$t3]
21:
22:    # Avanzar al siguiente elemento
23:    addi  $t0, $t0, 4      # Avanzar el puntero del array (4 bytes por palabra)
24:    addi  $t2, $t2, 1      # Incrementar el valor a almacenar
25:    addi  $t3, $t3, 1      # Incrementar el índice del array
26:
27:    # Volver al inicio del bucle
28:    j     fill_array
29:
30: end_fill:
31:    # Imprimir todo el array
32:    la    $t0, array      # Reiniciar el puntero del array a la posición base
33:    li    $t3, 0          # Reiniciar el índice del array a 0
34:
35: print_array:
36:    # Condición de salida del bucle
37:    beq   $t3, $t1, end_print # Si $t3 == $t1, salir del bucle
38:
39:    # Cargar el valor actual del array
40:    lw    $a0, 0($t0)      # Cargar el valor de array[$t3] en $a0
41:
42:    # Imprimir el valor
43:    li    $v0, 1           # Código de syscall para imprimir entero
44:    syscall                      # Llamar a la syscall
45:
46:    # Imprimir un espacio
47:    li    $v0, 4           # Código de syscall para imprimir cadena
48:    la    $a0, space       # Cargar la dirección del espacio en $a0
49:    syscall                      # Llamar a la syscall
50:
```

```
51:      # Avanzar al siguiente elemento
52:      addi $t0, $t0, 4      # Avanzar el puntero del array (4 bytes por palabra)
53:      addi $t3, $t3, 1      # Incrementar el índice del array
54:
55:      # Volver al inicio del bucle
56:      j      print_array
57:
58: end_print:
59:      # Fin del programa
60:      li     $v0, 10         # Código de syscall para salir del programa
61:      syscall                # Llamar a la syscall
62:
63:
```