

```

Procedimiento cargarvector(V, N)
[Tvector: V parámetro por referencia]
[entero: N parámetro por referencia]
variable
    entero: i
inicio
    leer N
    para i desde 1 hasta N hacer
        leer V(i)
    finpara
fin [cargarvector]

```

```

Procedimiento cargarvector2(V, N)
[Tvector: V parámetro por referencia]
[entero: N parámetro por referencia]
variable
    entero: i
    carácter: op
inicio
    i ← 0
    Repetir
        i ← i + 1
        leer V(i)
        escribir 'mas datos S/N'
        leer op
    Hasta que op = 'N'
    N ← i
fin [cargarvector2]

```

```

Procedimiento mostrarvector(V, N)
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
variable
    entero: i
inicio
    para i desde 1 hasta N hacer
        escribir V(i)
    finpara
fin [mostrarvector]

```

```

funcion buscar (V, N, K): entero
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
[del tipo del vector: K parámetro por valor]
variable
    entero: i, pos
inicio
    pos ← 0
    para i desde 1 hasta N hacer
        si V(i) = K
            entonces
                pos ← i
        fin
    finpara
    buscar ← pos
fin [buscar]

```

```

funcion buscarM (V, N, K): entero
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
[del tipo del vector: K parámetro por valor]
variable
    entero: i, pos
    lógico: encontrado
inicio
    pos ← 0
    i ← 1

```

```

    encontrado ← f
    mientras (i ≤ N) Y No encontrado hacer
        si V(i) = K
            entonces
                encontrado ← v
                pos ← i
            sino
                i ← i + 1
        fin
    finmientras
    buscarM ← pos
fin [buscarM]

```

```

Procedimiento modificar(V, N, X, pos)
[Tvector: V parámetro por referencia]
[entero: N parámetro por valor]
[del tipo del vector: X parámetro por valor]
[entero: pos parámetro por valor]
inicio
    V(pos) ← X
fin [modificar]

```

```

Procedimiento agregar(V, N, X)
[Tvector: V parámetro por referencia]
[entero: N parámetro por referencia]
[del tipo del vector: X parámetro por valor]
inicio
    N ← N + 1
    V(N) ← X
fin [agregar]

```

```

Procedimiento insertar(V, N, X, pos)
[Tvector: V parámetro por referencia]
[entero: N parámetro por referencia]
[del tipo del vector: X parámetro por valor]
[entero: pos parámetro por valor]
variable
    entero: i
inicio
    para i desde N hasta pos con paso -1 hacer
        V(i+1) ← V(i)
    finpara
    V(pos) ← X
    N ← N + 1
fin [insertar]

```

```

Procedimiento eliminar(V, N, pos)
[Tvector: V parámetro por referencia]
[entero: N parámetro por referencia]
[entero: pos parámetro por valor]
variable
    entero: i
inicio
    para i desde pos hasta N-1 hacer
        V(i) ← V(i+1)
    finpara
    N ← N - 1
fin [eliminar]

```

```

funcion buscarPar (V, N): entero
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
variable
    entero: i, pos, R
inicio
    pos ← 0

```

```
para i desde 1 hasta N hacer
    R ← V(i)-V(i)/2*2 // o bien R ← V(i)
mod 2
    si R = 0
        entonces
            pos ← i
    fin si
fin para
buscarPar ← pos
fin [buscarPar]

funcion buscarPar (V, N): entero
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
variable
    entero: i, pos, R
    lógico: encontrado
inicio
    pos ← 0
    encontrado ← f
    i ← 1
    Mientras (i ≤ N) Y No encontrado hacer
        si Resto (V(i), 2) = 0 // o bien V(i) mod
2
            entonces
                pos ← i
                encontrado ← v
            sino
                i ← i + 1
    fin si
Finmientras
buscarPar ← pos
fin [buscarPar]

funcion buscarRepetido (V, N): entero
[Tvector: V parámetro por valor]
[entero: N parámetro por valor]
variable
    entero: i, pos, R
inicio
    pos ← 0
    para i desde 1 hasta N-1 hacer
        para j desde i+1 hasta N hacer
            si V(i) = V(j)
                entonces
                    pos ← i
            fin si
        fin para
    fin para
    buscarRepetido ← pos
fin [buscarRepetido]
```