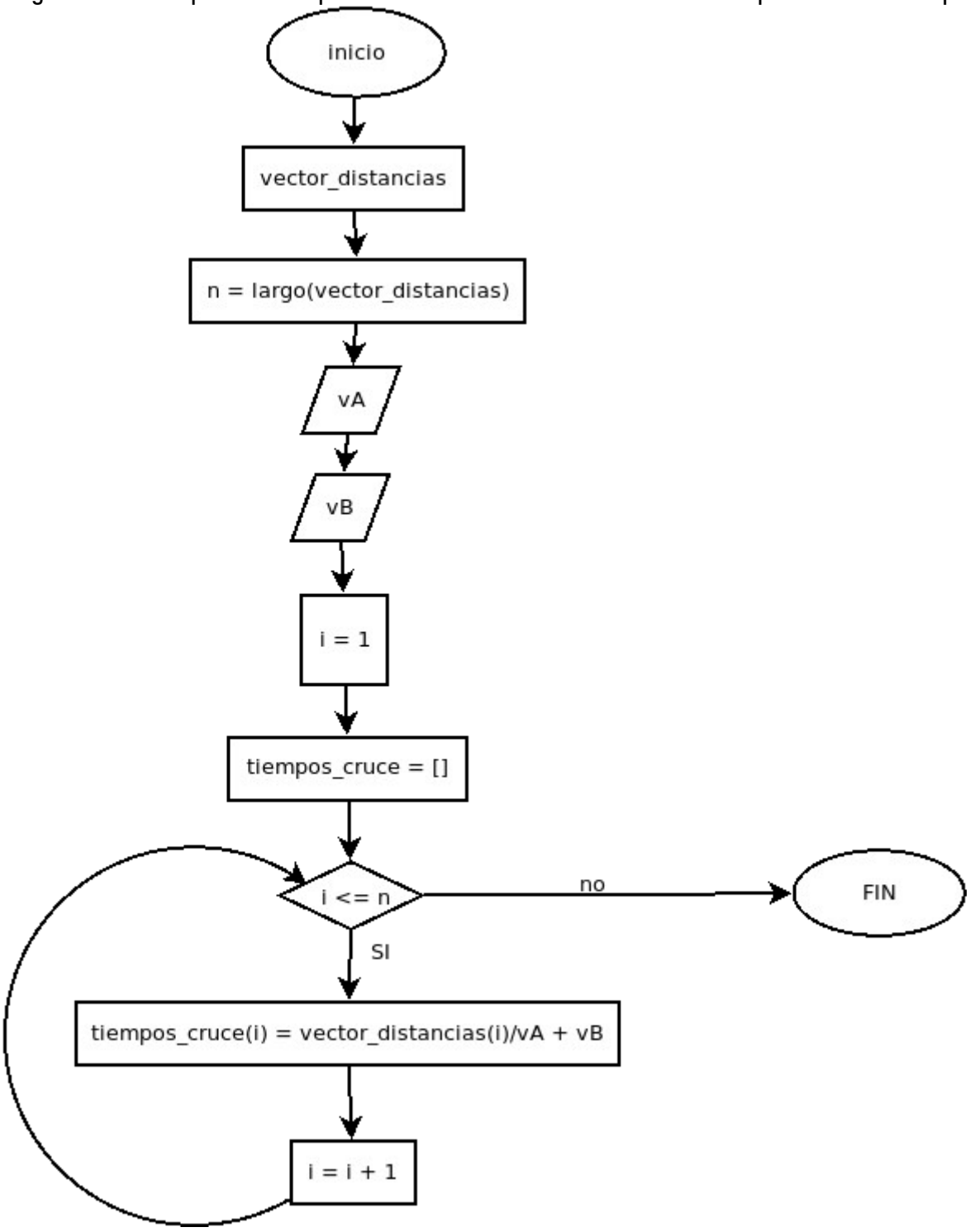
Primer Parcial – Computación1 Nombre:_____ Cédula:_____

Ejercicio 1 (7 puntos)

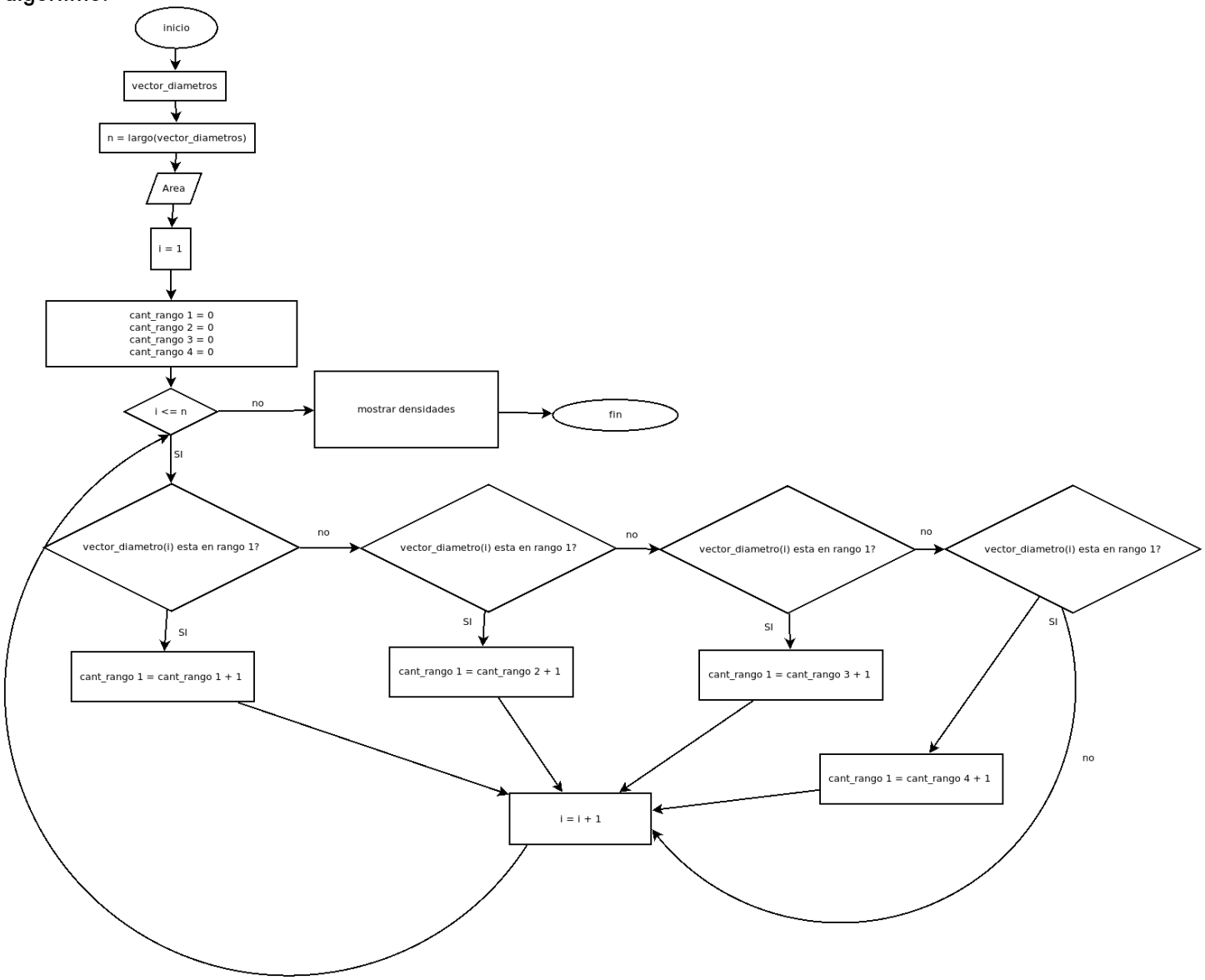
Dado un vector de distancias (como en la Tarea 1 y 2) dibujar un diagrama de flujo que represente un algoritmo que calcule el vector de los tiempos de cruce de dos camiones. Las velocidades de cada camión son datos que deben ser leídos en el algoritmo. Debe quedar claro para cada distancia como se calcula el tiempo de cruce correspondiente.



Primer Parcial – Computación1 Nombre:_____ Cédula:_____

Ejercicio 2 (7 puntos)

Dado la superficie de un área forestal y un vector con diámetros de arboles (como en la Tarea 2) dibujar un diagrama de flujo que represente un algoritmo que calcule el vector de densidades para 4 categorías de diámetro (10-20 cm, 20-30 cm, 30-40 cm y 40 cm y mas). La superficie en cuestión es un dato que debe ser leído por el algoritmo.



Ejercicio 3 (7 puntos)
Complete el código siguiente en Octave para saber si un numero X es un numero perfecto. Un numero perfecto es numero natural que es igual a la suma de sus divisores propios, ejemplo 6 = 1 + 2 + 3.

```
function resultado = es_perfecto(X)

  % Inicializamos la variable que almacenará la suma de los divisores
  suma = 0

  % Recorremos los divisores de X
  for i = 2:X
  % Si i es un divisor de X, lo sumamos a la suma
    if X %% i == 0
      suma += i
    end
  end

  % Si la suma es igual a X, entonces X es un número perfecto y devolvemos 1
  return suma == X

endfunction
```

Ejercicio 4 (7 puntos)
Rescriba el siguiente código donde se utilice solo la estructura de iteración **for**

```
v = [3, 5, 7, 9, 5, 13, 17, 19, 5, 27]
i = 1;
j = length(v);
X = 5;
lugares = [];

while i <= j
  if v(i) == X
     lugares = [lugares i]
  endif
  if v(j) == X
     lugares = [lugares j]
  endif
  i = i + 1;
  j = j - 1;
endwhile
```

```
v = [3, 5, 7, 9, 5, 13, 17, 19, 5, 27]
X = 5;
lugares = [];
j = 0;

for i = 1:length(v)/2
  if v(i) == X
    lugares = [lugares i]
  endif
  j = length(v)+1-i;
  if v(j) == X
    lugares = [lugares j]
  endif
endfor
```

Ejercicio 5 (7 puntos)
Describa con sus palabras que implementa esta función de Octave

```
function y = f(inputVector)

    inputLength = length(inputVector);

    y=1;
    for i = 1 : floor(inputLength / 2)
        if inputVector(i) ~= inputVector(inputLength + (1 - i))
            y = 0;
        end
    end
 end
```

La función comprueba si un vector es “capicua” (si se lee tanto en un sentido como del otro, tiene el mismo contenido en el mismo orden).