

Programación I

Guía de Errores de Diagrama Parte 3.



Índice

Objetivos:.....	2
Contenidos:	2
Errores en diagramas de flujo. Parte 3.	3
Ejemplo 1.	4
Ejemplo 2.	6
Ejemplo 3.	9
Ejemplo 4.	11
Ejemplo 5.	13
Ejemplo 6.	15
Ejemplo 7.	17
Ejemplo 8.	18
Ejemplo 9.	23
Bibliografía.....	27



Objetivos:

- Proporcionar ejemplos de errores comunes a la hora de realizar diagramas de flujo.

Contenidos:

- Ejemplos prácticos.



Errores en diagramas de flujo. Parte 3.

En el presente documento vamos a indicar una serie de diagramas con errores que hemos detectado a lo largo del tiempo y que resumimos aquí con la intención de no verlos repetidos en futuras correcciones.

Es importante recordar que nuestros diagramas son el paso previo a la codificación y que los diagramas están pensados para respetar ciertas formalidades, si las mismas no se cumplen tendremos dos problemas.

El primero es que los docentes al corregir los diagramas los consideraremos como erróneos, pues la corrección se realizara siguiendo esas formalidades.

El segundo, y más importante, es que, si el diagrama tiene errores y se lo codifica de esa forma, el programa no funcionará. El compilador es implacable al revisar la sintaxis de nuestros programas.

A continuación, presentaremos una serie de figuras y el motivo por el cual las mismas no son correctas.

Ejemplo 1.

Enunciado 1. Hacer un programa para ingresar una lista de 10 números, luego informar al final cuantos son positivos.

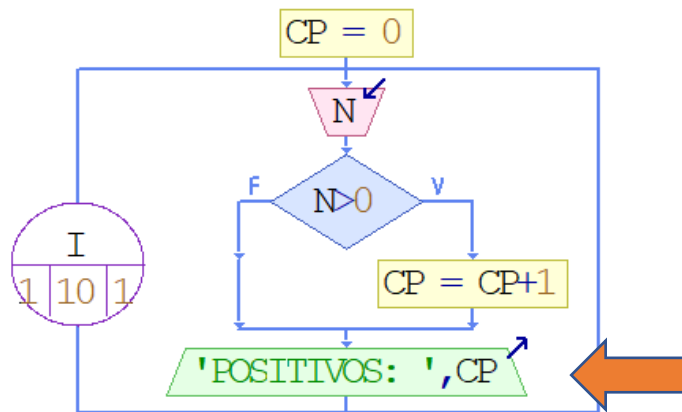


Figura 1a incorrecta

La figura 1a contiene un error muy habitual, y es el incluir el cartel de salida dentro del ciclo FOR. Cuando el enunciado dice “luego informar al final” lo que debemos entender es que ese cartel debe emitirse al final del programa y una ÚNICA VEZ.

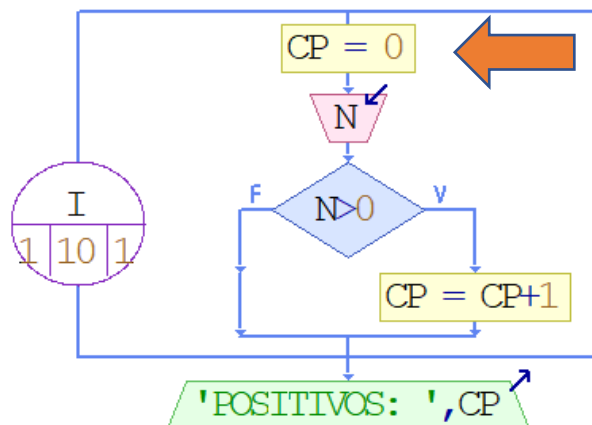


Figura 1b incorrecta

La figura 1b contiene otro error habitual y es confundir la ubicación de donde colocamos en cero un contador, en este caso el contador se llama CP. Si lo coloca dentro del FOR ese

contador vuelve a cero una y otra vez, y cuando muestre el total al final va a mostrar un total erróneo.

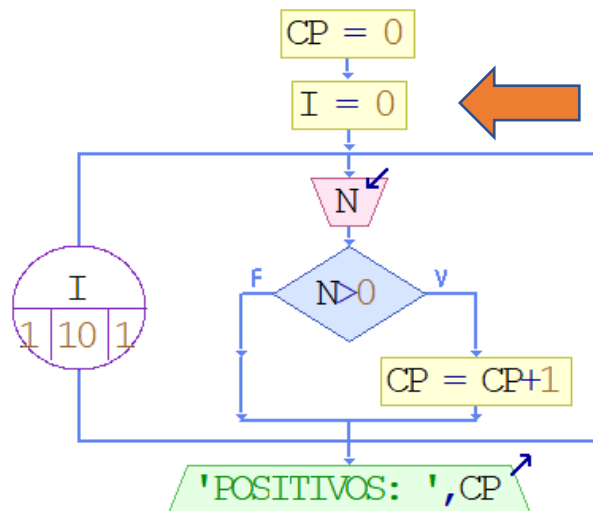


Figura 1c incorrecta

La figura 1C contiene un error no tan habitual pero no por eso poco importante. Una variable que se usa en un ciclo FOR ya tiene su propio valor inicial, en este caso $i = 1$, por lo tanto, la instrucción previa de darle valor $i = 0$ es ABSOLUTAMENTE INNECESARIA.

La siguiente figura es la solución correcta para el Enunciado 1.

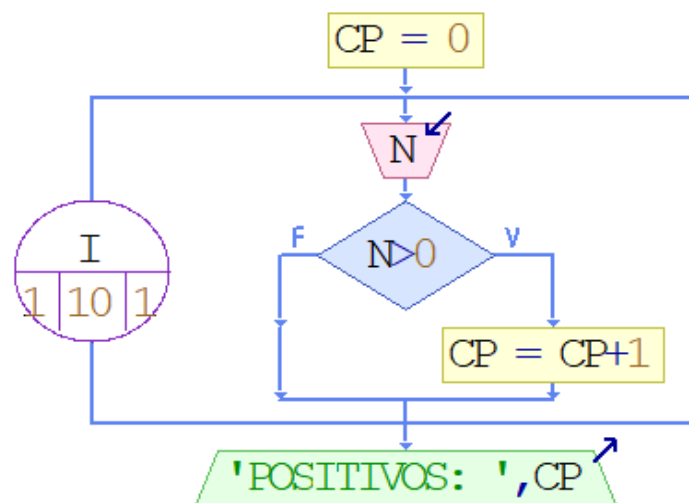


Figura 1d correcta

Ejemplo 2.

Enunciado 2. Hacer un programa para ingresar un número y luego emitir UN SOLO CARTEL que diga si el número es PRIMO o NO ES PRIMO.

La definición dice que un número es PRIMO cuando tiene SOLAMENTE 2 DIVISORES: el 1 y el número en sí mismo. Por ejemplo 13 es PRIMO (tiene SOLO 2 divisores 1 y 13), en cambio 6 NO ES PRIMO (tiene 4 divisores: 1, 2, 3 y 6).

Remarcamos una y otra vez la frase SOLO 2 DIVISORES porque algunos distorsionan la frase y la convierten en AL MENOS 2 DIVISORES, y eso es INCORRECTO. Si esa fuera la definición entonces el 6 también sería PRIMO, porque no solamente tiene 2 divisores, tiene 4, entonces no es PRIMO es SUPER PRIMO. Eso es una deformación de la definición.

Por lo tanto, cuando hacemos el diagrama para saber si un número es PRIMO debemos probar que en el rango desde el 1 hasta el valor de número haya SOLO 2 DIVISORES, y ese rango hay que probarlo completo.

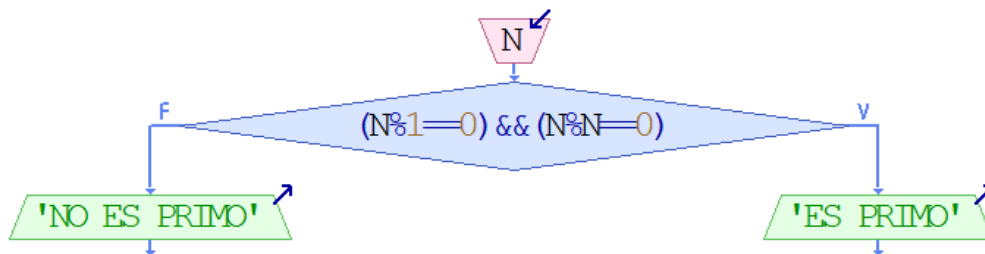


Figura 2a incorrecta

En este diagrama se toma la definición TEXTUAL de que un número es primo cuando tiene solamente como divisores al 1 y al número en sí mismo, pero se olvidan de hay que analizar también como divisores a los que están entre el 1 y el número en sí mismo.

Por otra parte, si analizamos lo que preguntamos en ese diagrama en el IF, SIEMPRE VA A DAR VERDADERO. TODO NUMERO ES MULTIPLO DE 1 Y DE SI MISMO. Es decir, sea lo que sea que valga N, ese IF siempre va a decir que “Es primo”, porque el análisis está mal hecho.

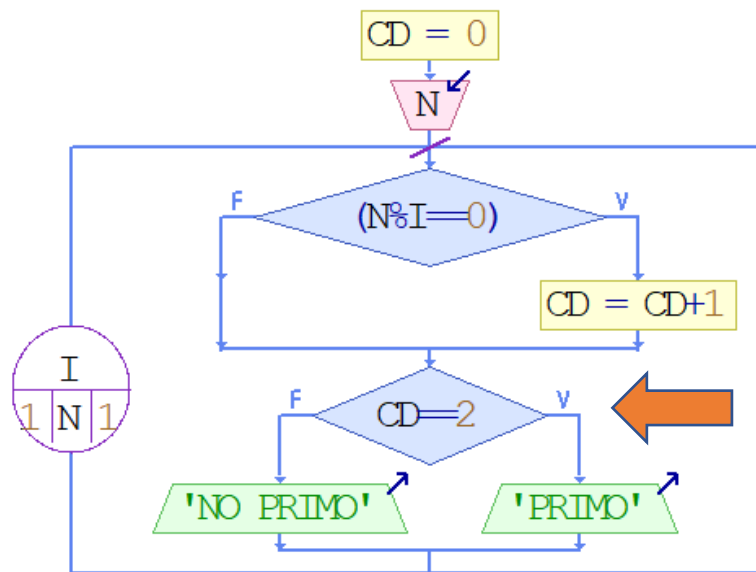


Figura 2b incorrecta

En la Figura 2B el error es colocar ese IF dentro del FOR, porque apenas le cuentas 2 divisores a un número te sale el cartel de “PRIMO”. La pregunta de cuantos son los divisores TIENE QUE IR FUERA DEL FOR, cuando recorriste TODO el rango desde el 1 hasta el valor de N.

Este diagrama tiene además otro error grave, si el cartel está dentro del ciclo FOR, va a salir repetidas veces. Si el diagrama lo pasamos a código C++ e ingresamos como número el 6, esto es lo que emite:

Vuelta i=1, sale el cartel: “NO PRIMO”.

Vuelta i=2, sale el cartel: “PRIMO”.

Vuelta i=3, sale el cartel: “NO PRIMO”.

Vuelta i=4, sale el cartel: “NO PRIMO”.

Vuelta i=5, sale el cartel: “NO PRIMO”.

Vuelta i=6, sale el cartel: “NO PRIMO”.

El usuario pidió UN CARTEL, el programa le devuelve 6.....

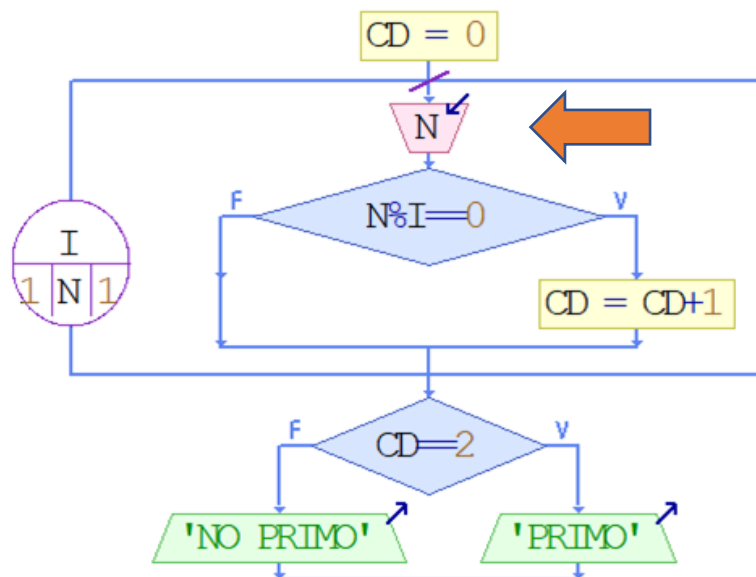


Figura 2c incorrecta

La Figura 2c tiene un GRAVE error que consiste en que el ciclo FOR se usa en este SOLAMENTE para generar y contar los posibles divisores de N, por lo tanto, no hay que leer varios números. El enunciado dice: “Hacer un programa para ingresar un número y luego...”. ¿En qué parte dice que hay que leer varios números? A veces la respuesta que recibimos cuando vemos este diagrama es: “Ah, yo vi que en los ejercicios de FOR la lectura va siempre dentro, ¿así que porque iba a pensar que en este caso era diferente...? Hay que recordar que en programación no hay que copiar y pegar diagramas. Cada caso tiene sus características.

La siguiente figura es la solución correcta para el Enunciado 2

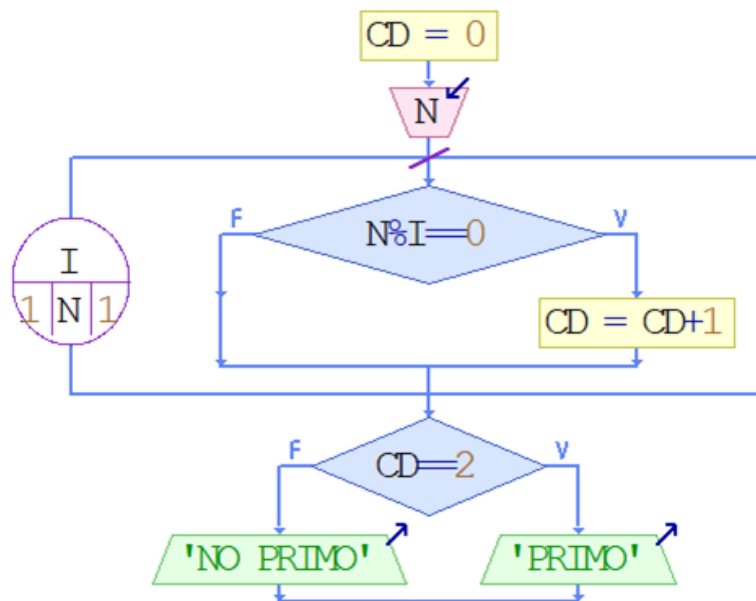


Figura 2d correcta

Ejemplo 3.

Enunciado 3. Hacer un programa para ingresar una lista de 10 números y luego emitir el máximo de la lista.

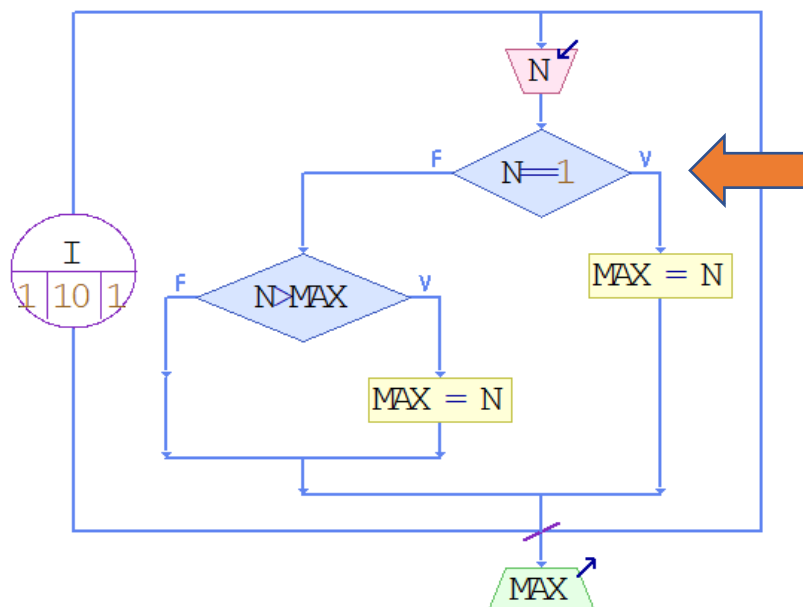


Figura 3a incorrecta

La Figura 3a es una de las varias formas para determinar si un número es el máximo de la lista, y consiste en detectar al primer número de la lista, guardarlo como MAX arbitrariamente, sin comparar, y al resto de los números de la lista, es decir del segundo en adelante, compararlos uno por uno y actualizar MAX si viene alguno mayor.

Pero la flecha indica un error que encontramos varias veces al corregir y que consiste en que le preguntan al número guardado en la variable N: “Vos sos el primer número de la lista?”. Y la variable N no tiene idea de esa pregunta. Su respuesta será: “Yo NO estoy acá para contar vueltas, yo recibo los números ingresados por el usuario. No puedo hacer dos cosas a la vez”. En este diagrama a quien se le debe preguntar si vamos por el primer ingreso, o primera vuelta, es a la variable del ciclo FOR, es decir la variable I.

Observe la figura 3b que resuelve de manera correcta el enunciado 3.

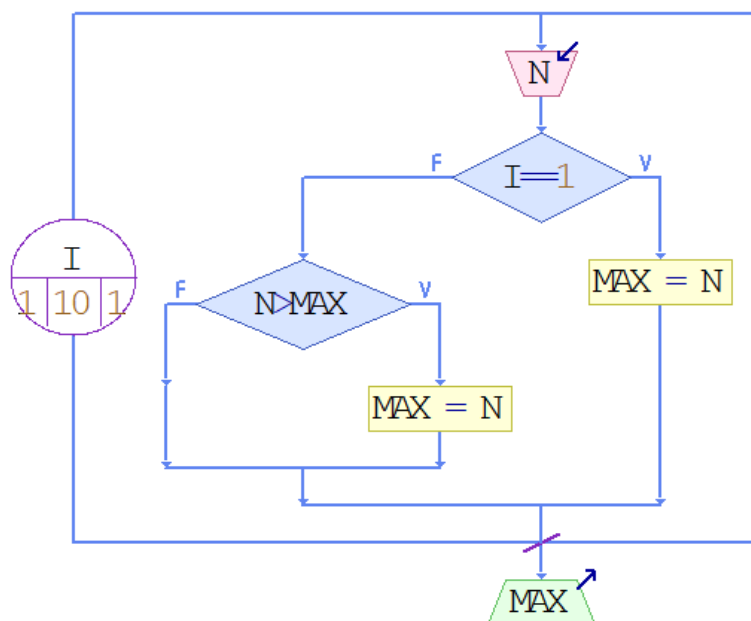


Figura 3b correcta

Recordamos nuevamente que, para buscar el máximo de una lista, esta no es la única forma de resolverlo. Simplemente usamos esa forma en particular para marcar ese error puntual de confundir la pregunta con N y con I porque es un caso testigo que sirve para no repetir el error en otros problemas similares.

Ejemplo 4.

Enunciado 4. Hacer un programa para ingresar las notas de los 100 alumnos de una materia de la facultad. Los alumnos se encuentran separados en 3 cursos, y los datos se ingresan desordenados. Cada registro contiene la siguiente información:

- Número de curso (1, 2 o 3).
- Nota (1 a 10)

Se pide calcular la nota promedio de los 3 cursos y luego indicar el NUMERO DE CURSO con mayor nota promedio.

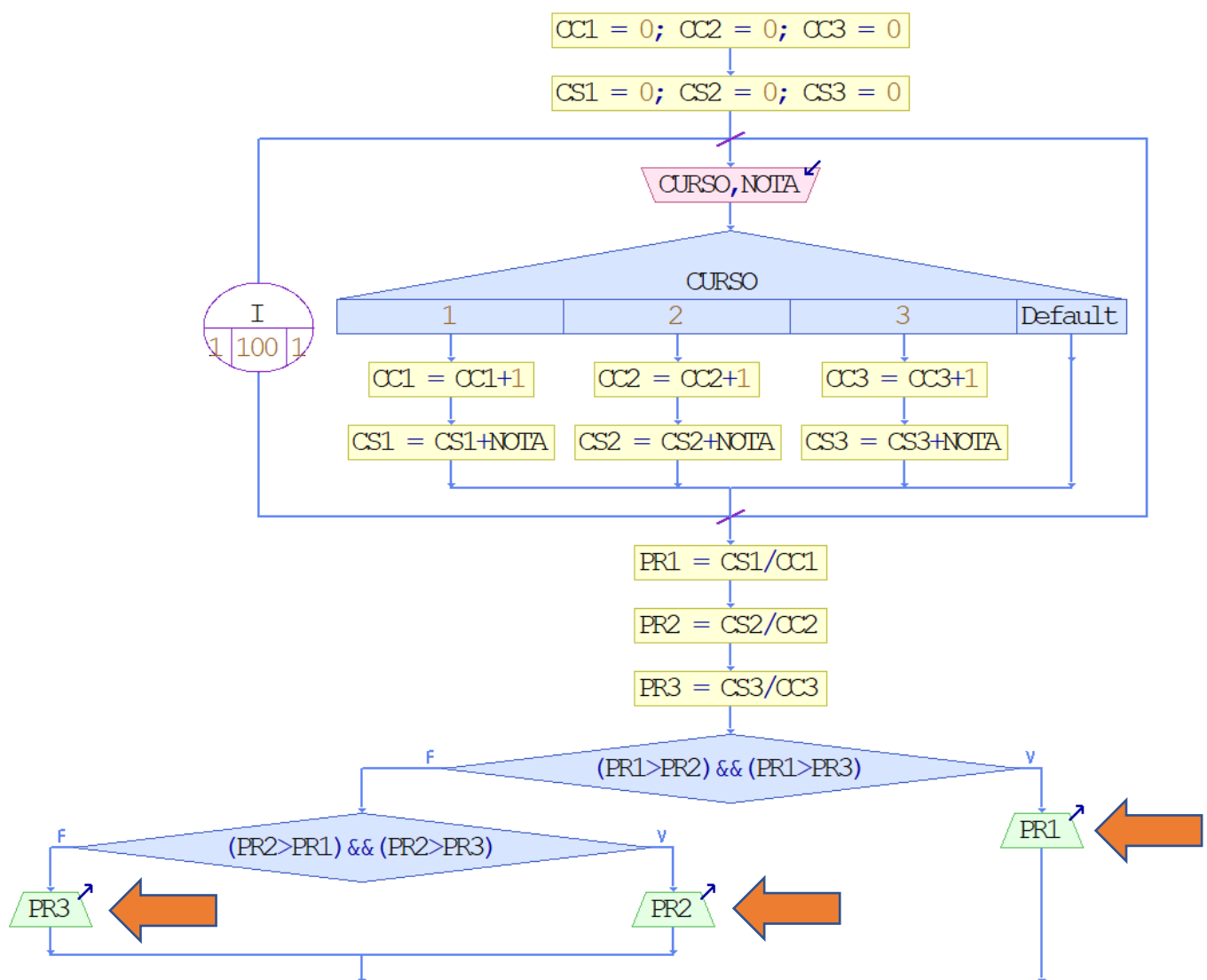


Figura 4a incorrecta

La Figura 4a tiene otro error común que encontramos al corregir, y es no entender la consigna de lo que nos pide el enunciado respecto a QUE DEBEMOS LISTAR al final del programa. El enunciado dice indicar “EL NUMERO DE CURSO” con mayor promedio. Para lograr ese objetivo tenemos varias alternativas, pero la más simple es un cartel que diga “Curso 1”, “Curso 2” o “Curso 3”.

Pero muchos asumen que listando el promedio del curso respectivo, que es lo que tienen las variables PR1, PR2 y PR3, ocurrirá “magia” y ese valor se convertirá en un cartel como los sugeridos anteriormente. Lamentablemente esa “magia” no va a ocurrir, y antes un caso como el siguiente, donde PR1=8, PR2=9.2 y PR3=7.5 el diagrama anterior listará a secas: 9.2. Entonces el usuario nos preguntará: “Perdón, pero CUAL ES EL CURSO 9.2?” Los cursos que hay son el 1, 2 y 3. No existe el curso 9.2.”

La consigna del enunciado no pide el MAYOR PROMEDIO, SINO CUAL CURSO LO TIENE. La siguiente Figura es correcta.

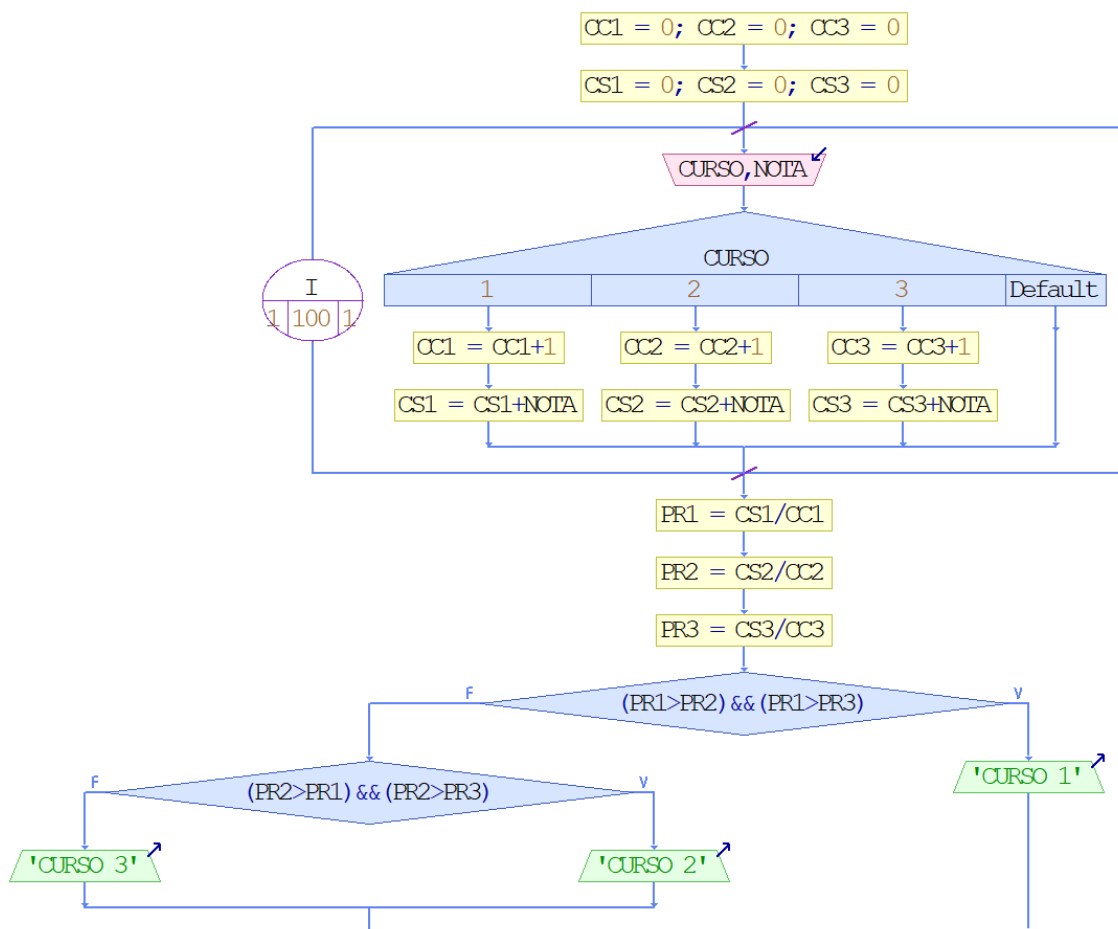


Figura 4b correcta

Otra solución válida es la siguiente:

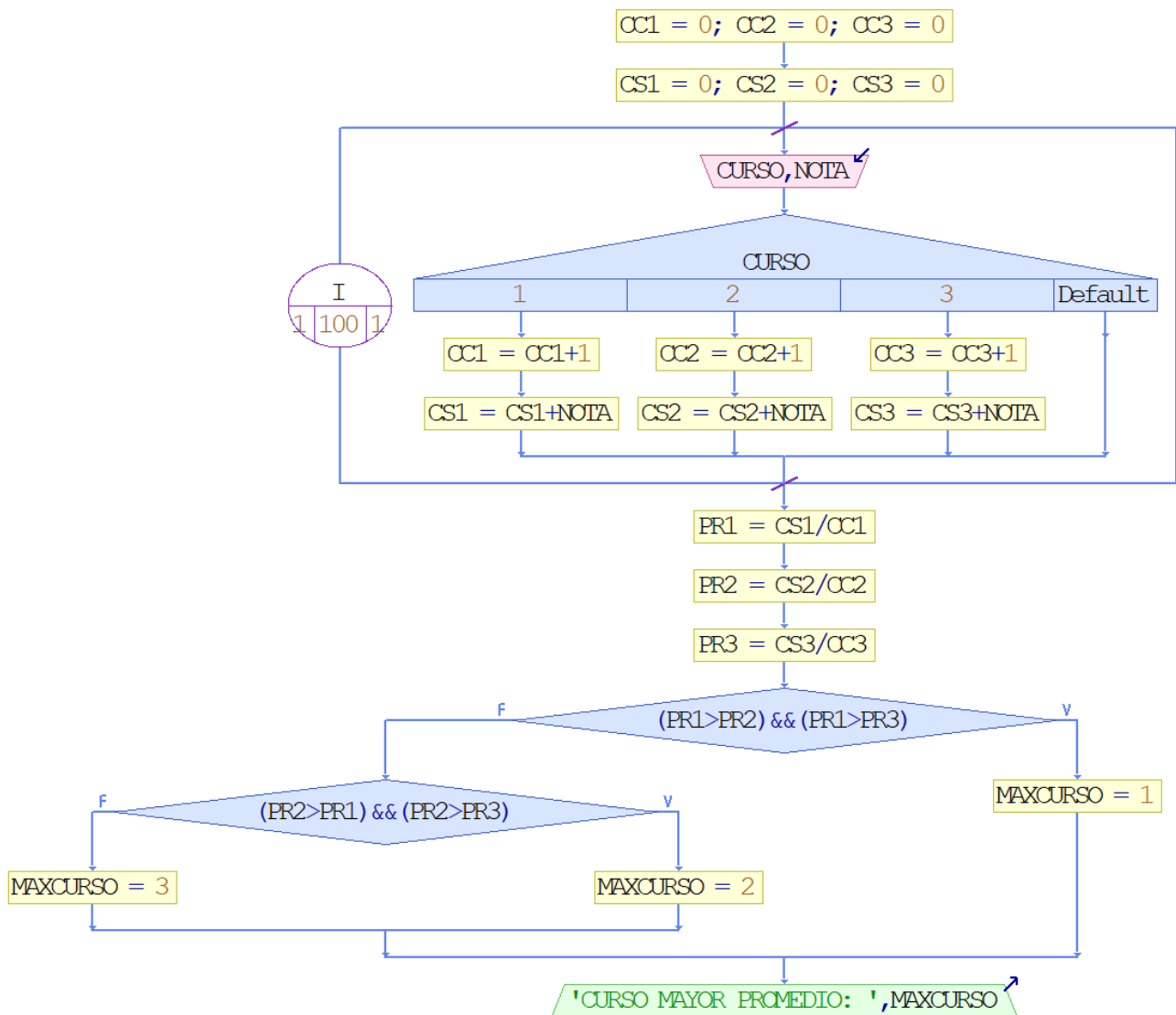


Figura 4c correcta

Ejemplo 5.

Enunciado 5. Hacer un programa para ingresar los precios de los 100 artículos. Cada registro contiene la siguiente información.

- Número de artículo (1 a 100).
- Precio Unitario.

Los datos no se ingresan ordenados. Se pide indicar al final el número del artículo que tiene el mayor precio unitario.

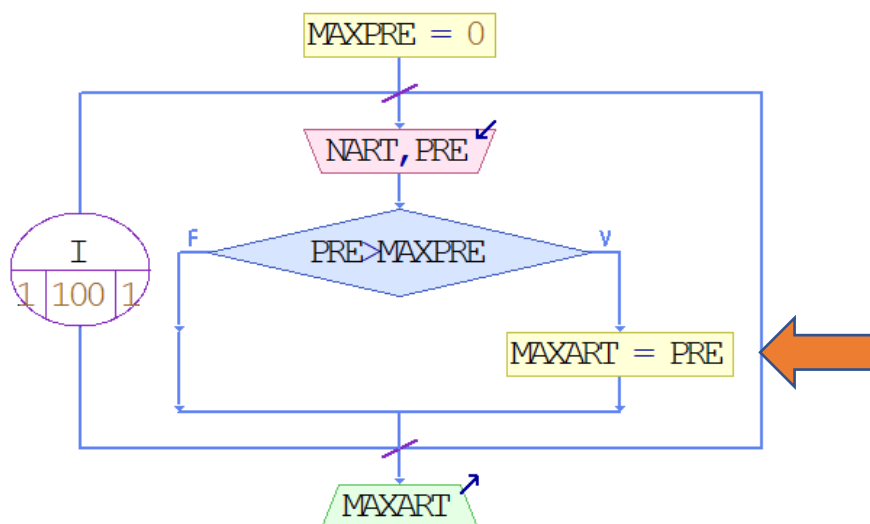


Figura 5a Incorrecta

La Figura 5a tiene un error muy común y es querer “ahorrar” variables y hacer una especie de 2 x 1 que lleva a errores. Si la variable MAXPRE la usamos en el programa para buscar cual es el precio máximo y la variable MAXART para guardar el número de artículo que tiene ese precio máximo, entonces CADA VARIABLE CUMPLE UNA FUNCION INDEPENDIENTE y NO hay que confundir una con la otra. Al momento que queremos guardar en la variable MAXART un Precio algo está fallando.

Probemos un caso simple donde los artículos fueran solamente tres.

Número de Artículo	Precio Unitario
2	100
1	150
3	120

Con esa muestra, si prueban el diagrama anterior, sale en el cartel que MAXART es.....120.

Ni 120 es el Precio más grande, ni 120 es número de artículo.

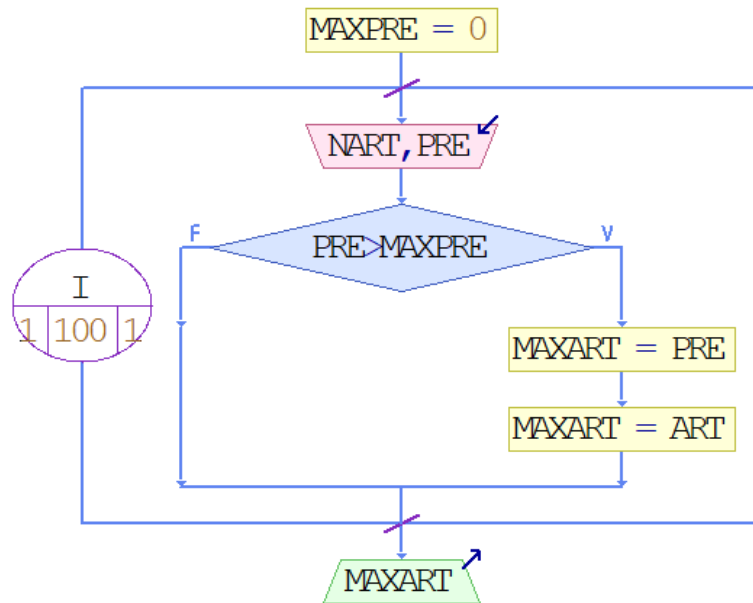


Figura 5b correcta

Si prueban los mismos datos que indicamos antes, sale el cartel que MAXART es 1.
¡Perfecto! Para cerrar el caso de este enunciado a veces escuchamos que para casos como este nos digan: “¿Pero si el MAXPRE no lo tenemos que listar, entonces para que lo usamos en el programa?”. Y la respuesta es: “El hecho de que ese dato no salga listado no significa que no te sirve. No hay forma de averiguar cuál es el artículo más caro, si no comparas los precios. Y para eso te sirve la variable MAXPRE. Independientemente de si luego la listas o no. Si siguiéramos esa lógica, la variable i del reloj también estaría de más, porque no sale listada al final...”.

Ejemplo 6.

Enunciado 6. Hacer un programa para ingresar las 200 ventas de un comercio que tiene 3 sucursales. Cada registro contiene la siguiente información.

- Número de sucursal (1, 2 o 3).
- Importe de la venta.

Los datos no se ingresan ordenados. Se pide indicar en que sucursal se recaudó mayor cantidad TOTAL de dinero.

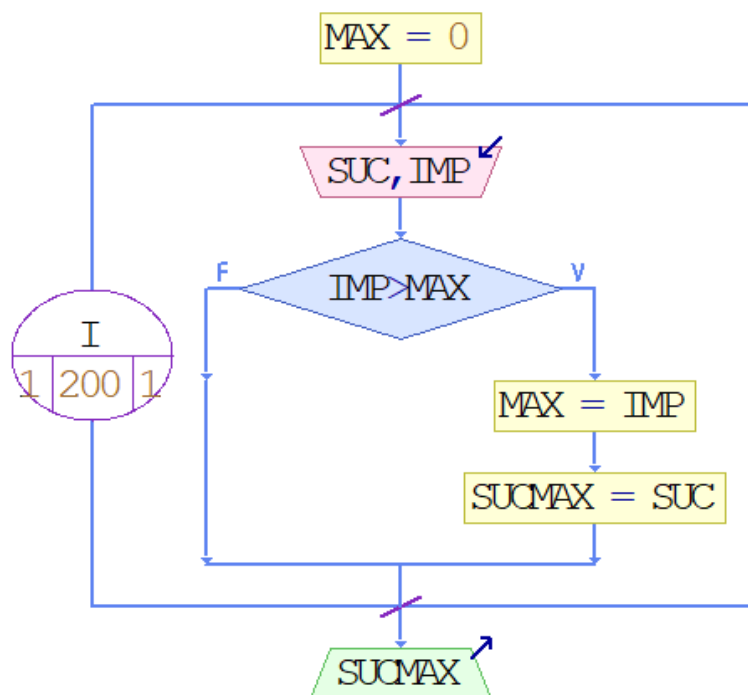


Figura 6a incorrecta

La Figura 6a tiene un error que en realidad es no comprender exactamente lo que el enunciado nos pide. Lo que en verdad hace ese diagrama es buscar la venta de mayor importe e indicar luego de que sucursal es esa venta individual. Pero eso no significa que esa sucursal sea la de mayor recaudación total, el hecho de que esa sucursal tenga una venta importante, no la convierte en la de mayor recaudación total.

Veamos este ejemplo:

Número de Sucursal	Importe de la venta
2	90
1	150
3	50
2	70
3	105

Si sumamos veremos que la sucursal 1 recaudó 150, la sucursal 2 recaudó 160 y la sucursal 3 recaudó 155. Es decir que la que mayor recaudó fue la sucursal 2, luego la 3 y luego la 1. Sin embargo, el diagrama anterior dice que la Sucursal de mayor recaudación es la 1.... Sale esa errónea conclusión por una venta que tuvo de 150.

A continuación, el diagrama que resuelve correctamente el enunciado 6.

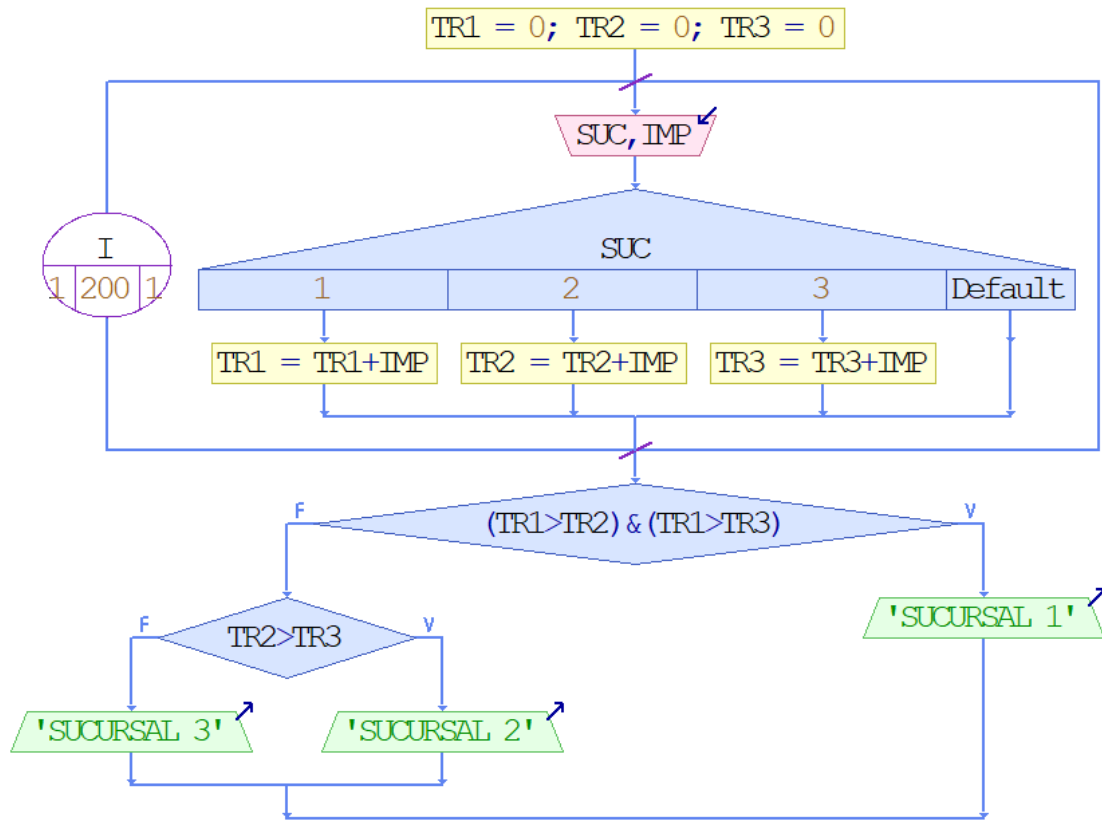


Figura 6b correcta

Ejemplo 7.

Enunciado 7. Hacer un programa para ingresar un número y listar por pantalla sus divisores.

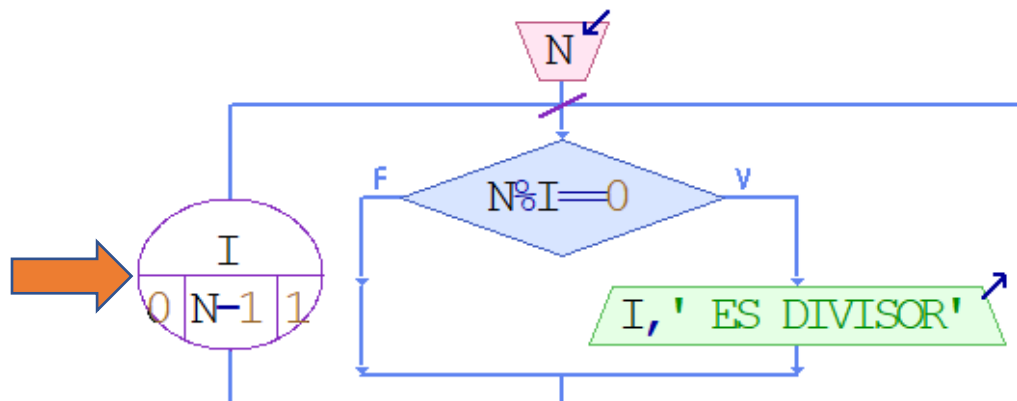


Figura 7a incorrecta

La Figura 7a muestra un error donde algunos consideran que para probar que divisores tiene un número el FOR tiene que ir desde 0 hasta N-1. Pero eso tiene un problema, si bien el FOR va a dar N vueltas, porque si por ejemplo N=6, el FOR ira de 0 hasta 5, y son 6 vueltas, pero cuando se intente probar en la primera vuelta si 0 es divisor de N va a dar error, y por otra parte no llegamos nunca a probar con el valor $i=6$, siendo que 6 es múltiplo de 6 y por lo tanto debería salir por pantalla. Entonces la cuestión de si poner un 1 o un 0 como valor inicial, o poner N o N-1 como tope, DEPENDE del caso. En este caso puntal hay que hacerlo de 1 a N.

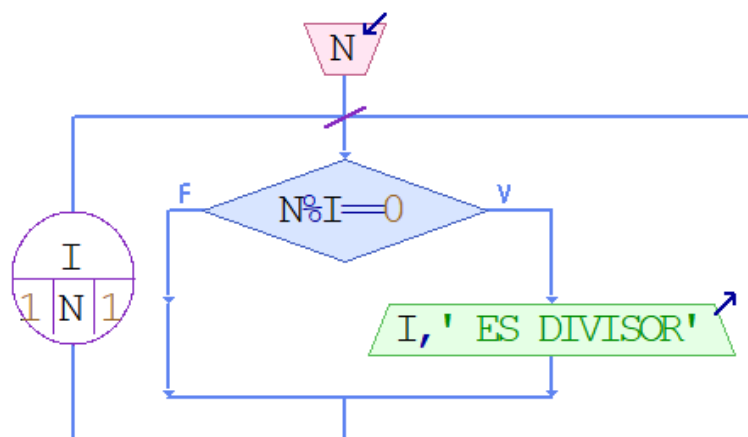


Figura 7b correcta

Ejemplo 8.

Enunciado 8. Hacer un programa para ingresar una lista de 5 números, luego informar el máximo de los múltiplos de 3.

Ejemplo A: 80, 12, 5, 9, 20. Se listará Máximo 12.

Ejemplo B: -5, -9, -12, -8, -15. Se listará Máximo -9.

Este enunciado y los ejemplos que damos no advierten de una situación importante, para muchos cuando en un ejercicio se habla de buscar un máximo, eso es sinónimo de positivo, y eso es un ERROR.

En el ejemplo B, surge que el máximo de los múltiplos de 3 de esa lista es -9. Entonces atento con eso y no confundir la cuestión de si buscamos un máximo, entonces seguro es positivo, y si buscamos un mínimo, entonces seguro es un negativo. ESA IDEA ES ERRONEA.

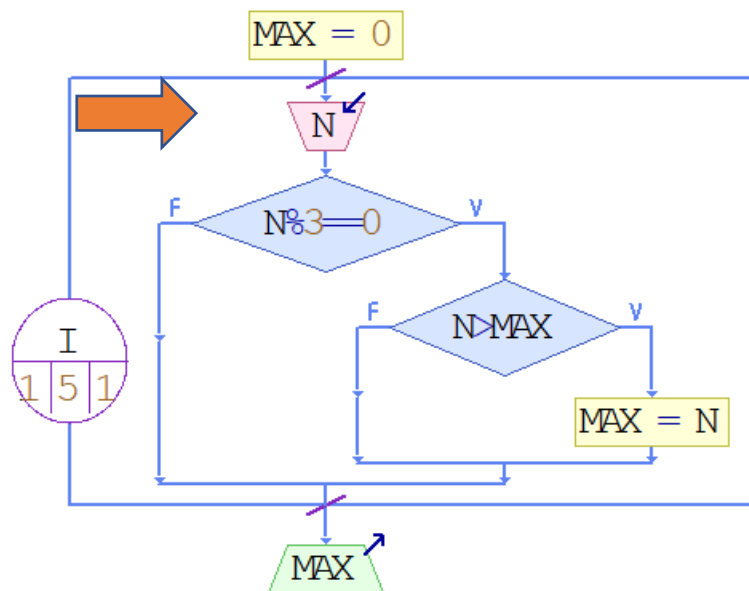


Figura 8a incorrecta

La figura 8A muestra el error de suponer que si buscamos un máximo entonces es un positivo, entonces que mejor que darle a MAXM3 el valor inicial 0.

Si se prueba ese diagrama con la lista Ejemplo B, el valor que sale al final como MAX es 0, porque como ninguno de los que son múltiplos de 3 lo superan, queda ese valor “de clavo”.

Explicado lo anterior, entonces queda claro que no es correcto darle valor inicial a MAX, para ir sobre seguro, conviene tomar un valor de la lista de los que el usuario ingresa para así evitar que quede un valor ficticio como sucede en la figura 8a con el valor 0. Que mejor idea entonces que leer el primer valor de la lista, guardarlo como MAX, y luego leer y comparar el resto.

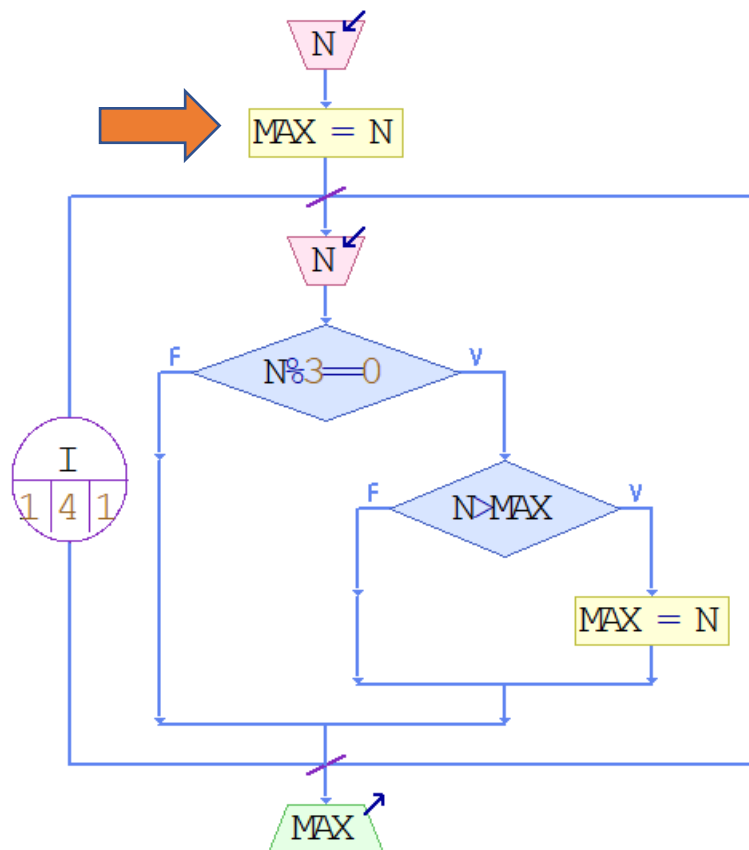


Figura 8b incorrecta

Otro diagrama que falla, fijense que pasa si probamos esta lista:

Ejemplo C: 80, 12, 5, 9, 20.

Con esa lista, y con ese diagrama, sale que el máximo múltiplo de 3 es el número 80.

PERO EL 80 NO ES MULTIPLIO DE 3. El error es cargarlo allí en la primera lectura como si lo fuera, y luego otra vez nos queda “de clavo”.

Entonces el error es que cuando leemos el primer número no analizamos si es múltiplo de 3, entonces ponemos un IF y listo. A ver veamos este diagrama.

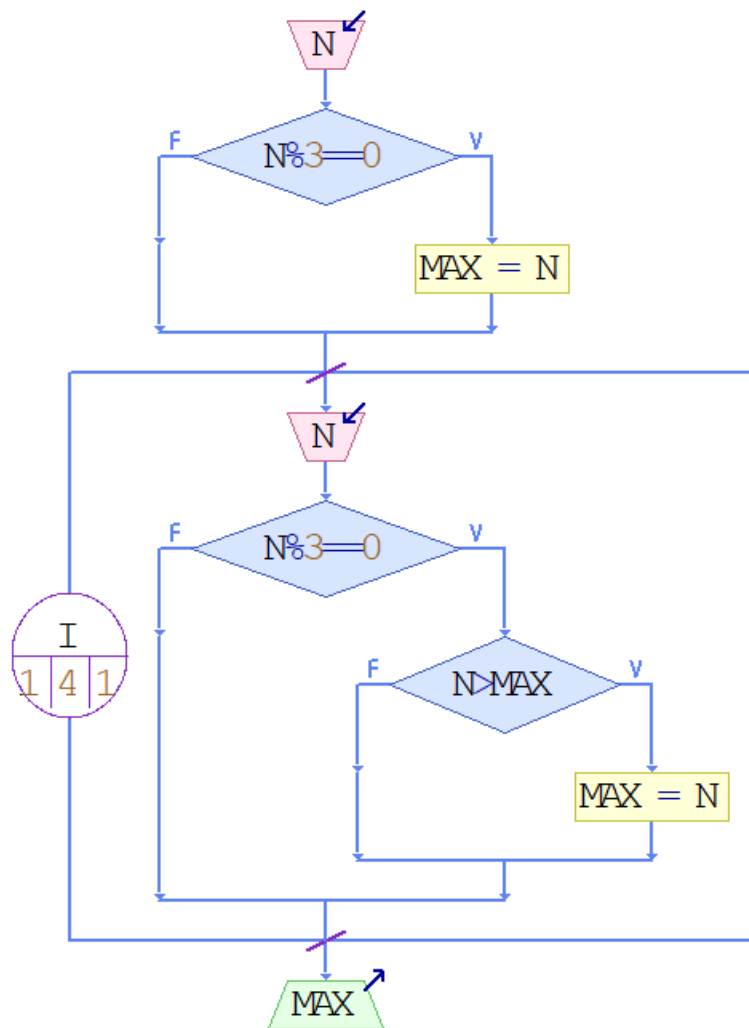


Figura 8c Incorrecta

Otro diagrama que falla. Si probamos la misma lista que antes,

Ejemplo C: 80, 12, 5, 9, 20.

Con esa lista cuando se lea el 80 y lo analicemos como múltiplo de 3 en el primer IF, eso va a ser falso, y entonces el valor de MAX va a valer BASURA.

Un valor BASURA en programación es impredecible, puede ser un valor grande o pequeño, negativo o positivo, etc. No podemos jugar con ellos. Si por caso el valor basura que MAX fuera 15125, no va a haber ninguno de lista que lo supere. Y cuando el usuario lo vea listado al final va a decir: “y que ese número, si yo NUNCA lo ingrese como parte de mi lista”.



Para resolver esta situación una posibilidad que tenemos es CONTAR a los múltiplos de 3 y cuando aparezca el primero de ellos, recién ahí lo guardamos como MAX, luego con los siguientes vamos comparando. NO PODEMOS DEPENDER NI ADIVINAR cuando viene ese primer múltiplo de 3.

Vean este caso de una lista de 10 números. 8, 20, 14, 7, 11, 13, -17, 19, 5, 12.

¿Cuándo aparece el “primer número múltiplo de 3”? Aparece en último lugar, en la ubicación 10. ¿Eso está mal? PARA NADA. No hay nada en el enunciado que te diga mira que la lista que vamos a cargar el primer múltiplo de 3 va a venir rapidito. Eso es pura fantasía.

A continuación, el diagrama que resuelve el Enunciado 8 correctamente.

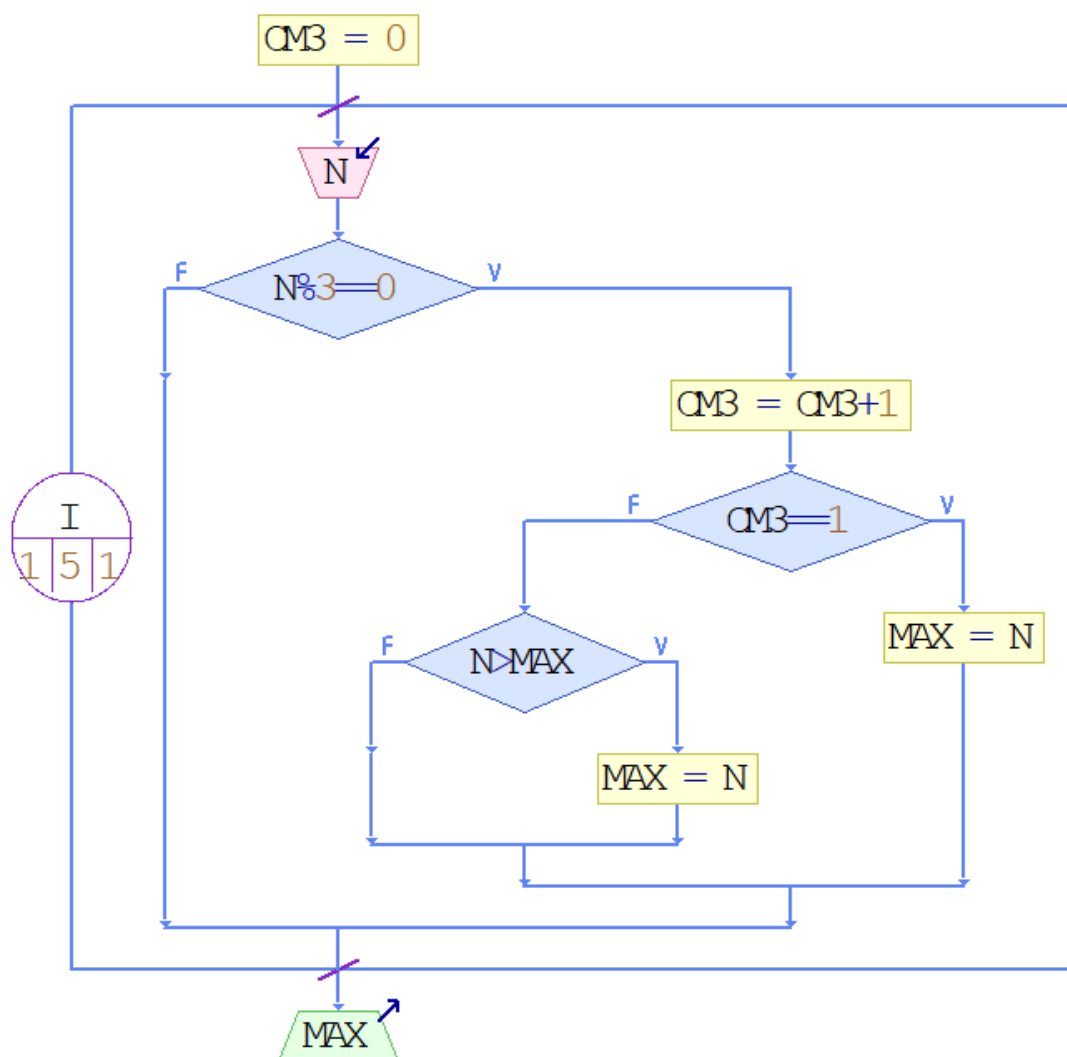


Figura 8d correcta



Ejemplo 9.

Enunciado 9. Hacer un programa para ingresar por teclado los saldos de los clientes de un banco que tiene tres sucursales. Son un total de 150 clientes y cada registro contiene los siguientes datos:

- Número de cliente (1 a 150).
- Numero de sucursal (1, 2 o 3).
- Saldo del cliente.

Los datos no se ingresan ordenados.

Se pide determinar e informar: El número de sucursal que tenga el mayor porcentaje de clientes con saldo superior a \$ 20.000.

Para comprender este ejemplo conviene hacer una lista de cómo pueden ingresar los datos.

Número de cliente	Número de sucursal	Saldo
120	3	15.000
7	1	50.000
8	3	85.000
12	2	25.000

Y así hasta completar 150 registros. De este ejemplo queda claro de que los datos NO SE INGRESAN ORDENADOS. No se ingresa el cliente 1, luego el cliente 2, etc.

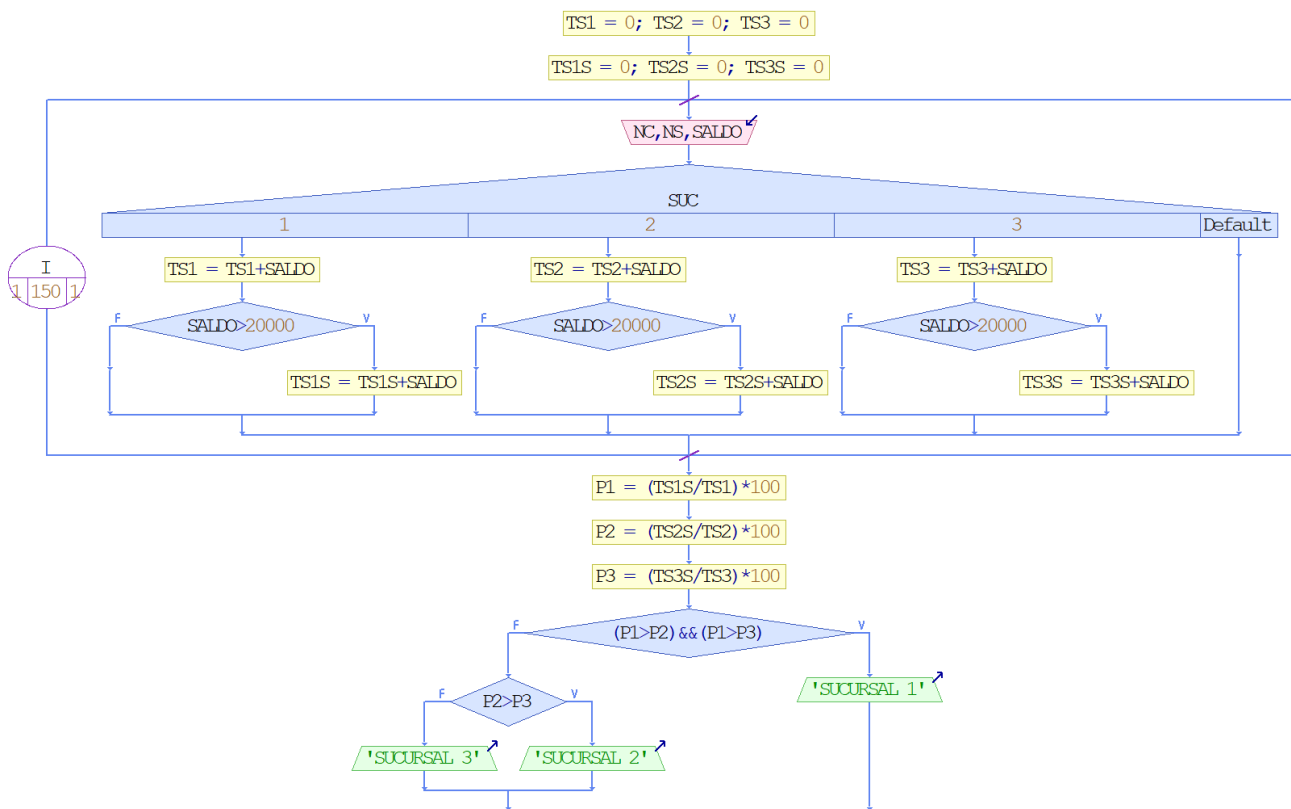


Figura 9a incorrecta

La Figura 9a parece perfecta, pero tiene un grave error, y consiste en no comprender el concepto de PORCENTAJES. Para calcular el porcentaje de clientes que tiene una sucursal con saldos mayores a 20.000 lo hay que hacer es CONTAR a los clientes, no SUMAR sus saldos. Esa suma de saldos no sirve de nada para calcular porcentajes.

Vamos a un ejemplo bien sencillo.

Caso 1. Tenemos una de las sucursales que tiene 4 clientes, de los cuales 3 clientes tienen saldo 30.000 y uno tiene saldo 10.000. ¿Qué porcentaje de clientes tiene saldo mayor a 20.000? Ese porcentaje es 75%, son 3 clientes de 4 en total.

Caso 2. Tenemos otra sucursal que tiene 2 clientes, de los cuales un cliente tiene saldo 1.000.00 y el otro tiene saldo 1.000. ¿Qué porcentaje de clientes tiene saldo mayor a 20.000? Ese porcentaje es 50%, es 1 cliente de 2 en total.



Es decir, de ambos casos el caso 1 tiene más porcentaje. Pero si cuando hacemos la cuenta, en lugar de CONTAR clientes, sumamos sus saldos, como en la figura 9a, vamos a tener este resultado para cada uno de los casos:

Caso 1, los clientes que tienen saldo mayor a 20.000, tiene cada uno 30.000 y son 3 clientes, suman 90.000 pesos, y el otro cliente tiene 10.000. Si calculamos el porcentaje siguiendo el diagrama, es $90.000/100.000 \times 100$, el resultado dice que la sucursal tiene 90% de clientes con saldo mayor a 20.000, ese porcentaje es incorrecto.

Caso 2, los clientes que tienen saldo mayor a 20.000, es uno solo y tiene 1.000.000 y el otro cliente tiene 1.000. Si calculamos el porcentaje siguiendo el diagrama, es $1.000.000/1.001.000 \times 100$, el resultado dice que la sucursal tiene 99,90%, nuevamente ese porcentaje es incorrecto.

Pero el error es que con ese cálculo la conclusión es que el caso 2 tiene más porcentaje de clientes con cuentas de saldo mayor a 20.000 que el caso 1. Y ESO ESTA MAL. El caso 2 tiene un cliente con un saldo muy grande y eso distorsiona todo porque si sumamos los saldos en lugar de contar clientes (que es lo que deberíamos haber hecho), cometemos un error y mezclamos dos cosas diferentes.

El siguiente diagrama si hace el cálculo correcto.

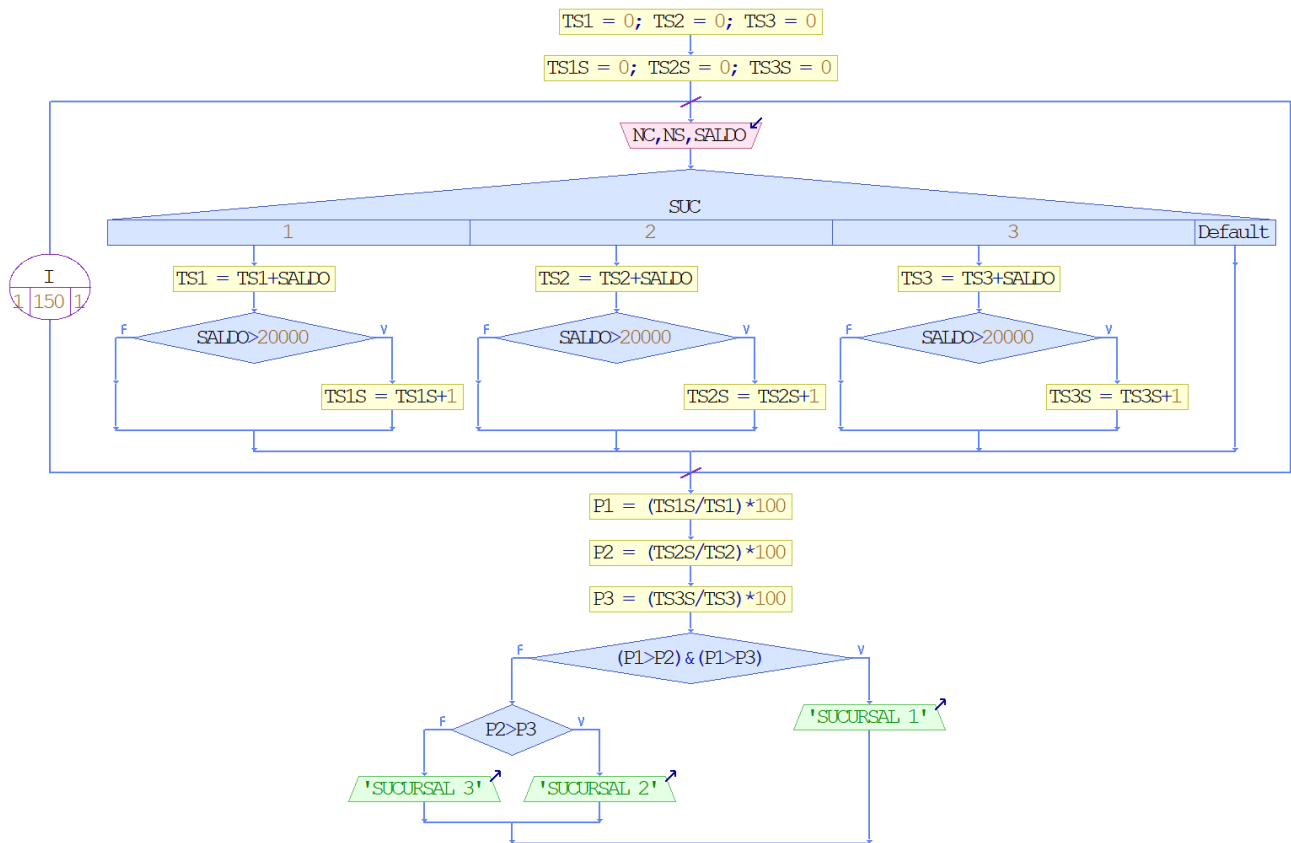


Figura 9b correcta



Bibliografía

- Creado por la cátedra y dirección de la carrera.
- Ejemplos preparados por el Ing. Claudio Fernández.