**Ejercicio1: Método del Camino básico. Pruebas estructurales Caja Blanca**

Próxima clase de Taller: uno o dos alumnos pueden presentar su caso del inicio de la clase (puntos base de solemne 1)

**INSTRUCCIONES GENERALES**

1. Desarrollar los casos de prueba (CP) de los caminos básicos establecidos en el ejercicio
2. Para ello desarrolle el grafo y los caminos básicos derivados de la Complejidad ciclomática calculada
3. Para profundizar en este método puedes ver los siguientes videos:

(1) https://www.youtube.com/watch?v=9N5vPeSWRfQ&t=92s

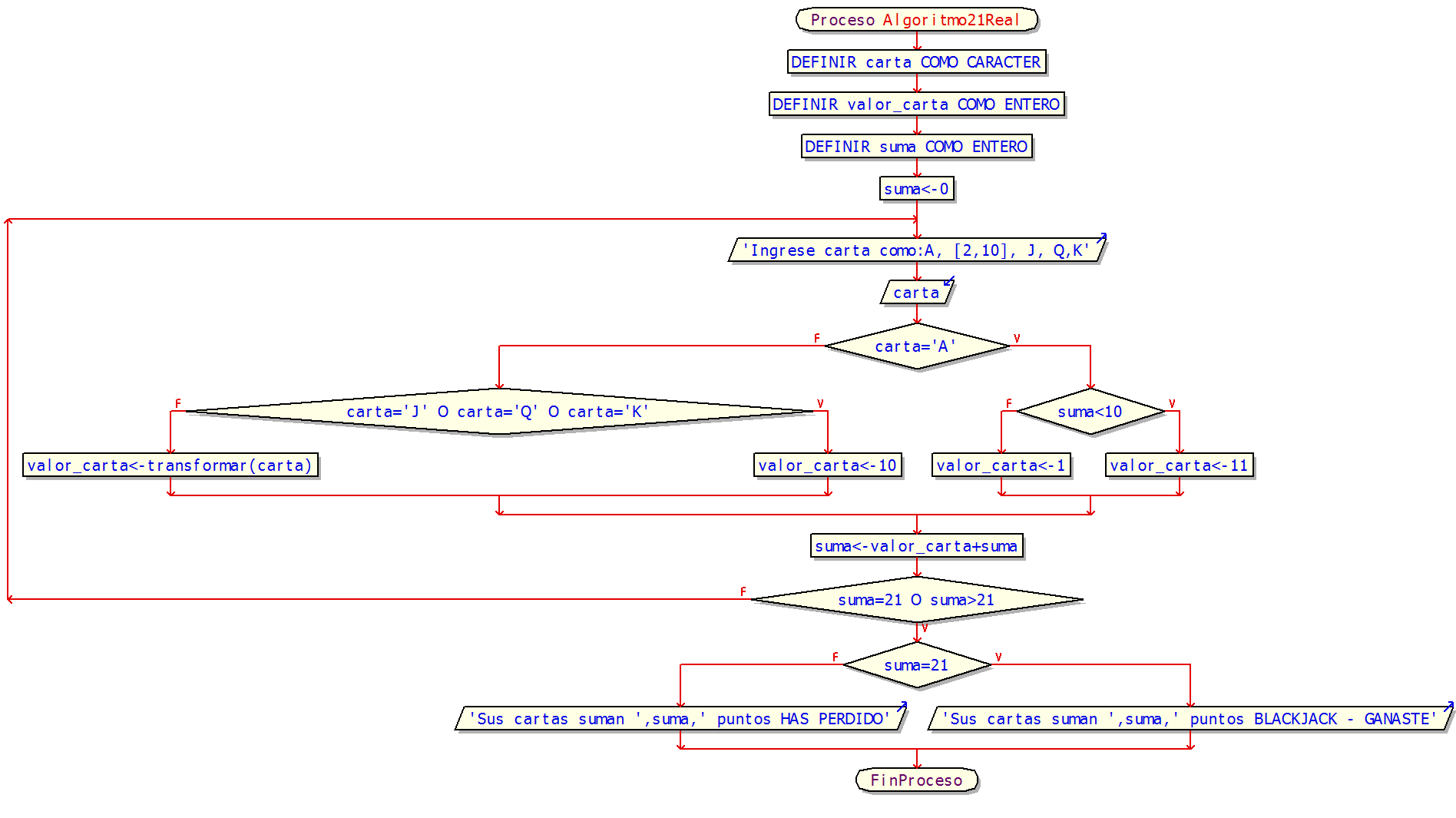
(2) https://www.youtube.com/watch?v=GVegCwwfBZ0

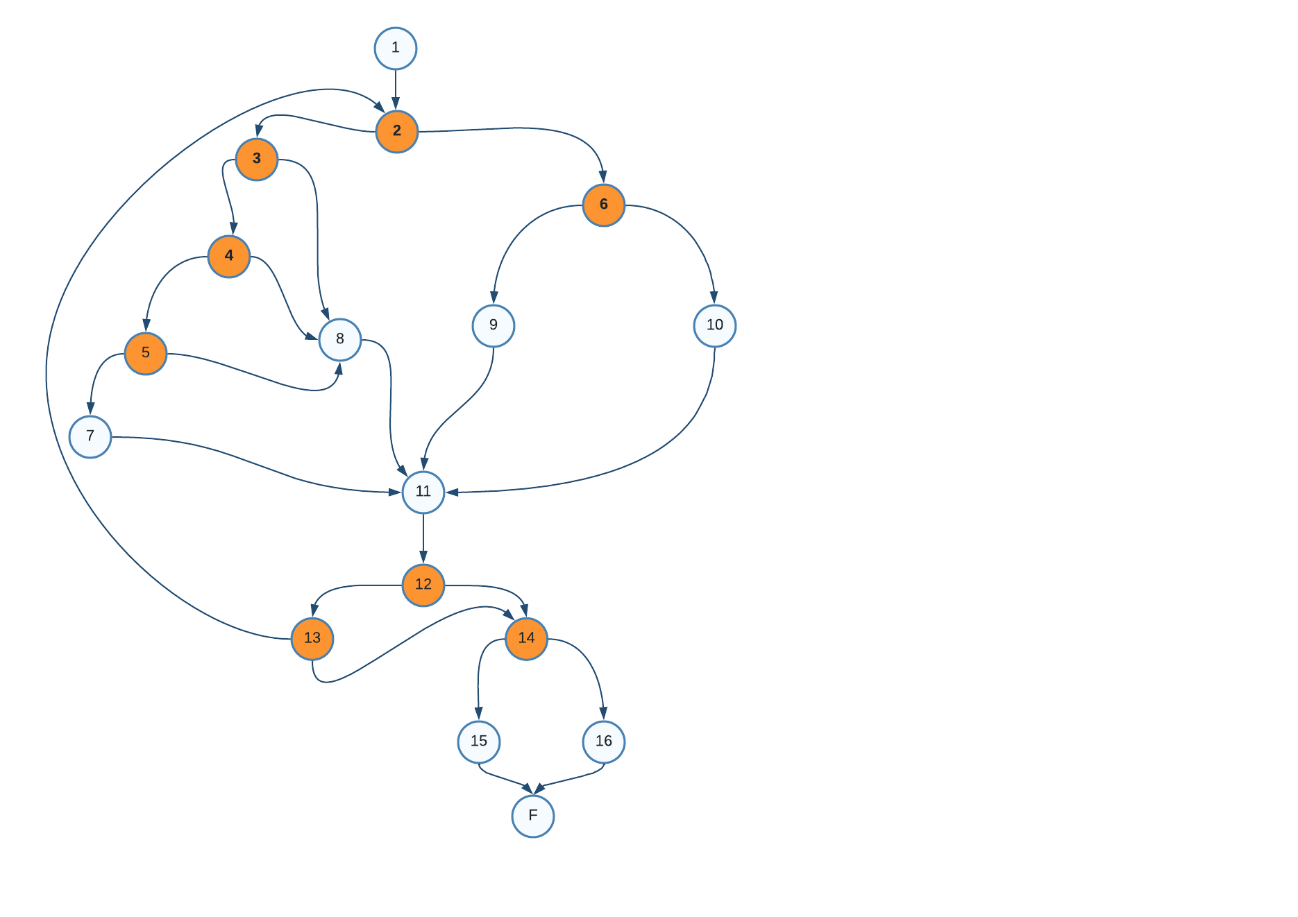
**CASO**

Calcule todos los caminos posibles (caminos básicos) que debe recorrer y probar la variable de entrada para el algoritmo de 21 real. El juego se termina cuando las cartas suman 21 o cuando son mayores que 21. Puede ingresar cartas aleatorias de 1 en 1.

**Especificación del Algoritmo 21 Real Solución Propuesta DESARROLLADA**

**Supuestos:**

* Las cartas J,Q y K valen 10 puntos
* La carta As, puede valer 1 u 11 según convenga
* La carta As debe tomar valor al ser evaluada y toma valor con respecto a la suma a las cartas que ya aparecieron.
* La carta As cuando es la primera toma el valor más alto.
* Imagen1: Algoritmo en PSeInt del 21 Real.



**1. Gráfica del flujo de regiones del algorítmico 21 Real. Define las regiones de la Complejidad ciclomática.**

**2. Determinar el valor de la complejidad ciclomática V(G)**

* V(G), contar el número de regiones
* V(G)=E – N +2 donde E es el número de aristas y N es el número de nodos

V(G)= P+1, donde P es la cantidad de nodos predicados

V(G) = 24-17 +2= 9

V(G)= 8+1 =9

**3. Determinar el conjunto básico de rutas linealmente independientes (deben aportar alguna arista nueva)**

**C1: 1,2,3,4,5,7,11,12,13,2**

**C2: 1,2,6, 9, 11, 12,13,2**

**C3: 1,2,6,10, 11, 12,13,2**

**C4: 1,2,3,8, 11,12,13,2**

**C5: 1,2,3,4,8, 11,12,13,2**

**C6: 1,2,3,4,5, 8, 11,12,13,2**

**C7: 1,2,3,4,5,7,11,12, 14, 15, F**

**C8: 1,2,3,4,5,7,11,12, 14, 16, F**

**C9: 1,2,3,4,5,7,11,12, 13, 14, 15 F**

**4. Desarrolla los Casos de Prueba del Caso. Desarrolla todos los casos de prueba y las salidas del programa**

|  |  |  |
| --- | --- | --- |
| **CAMINO** | **ENTRADAS** | **SALIDA** |
| **1,2,3,4,5,7,11,12,13,2** | **Carta=6, suma =0, Suma = 6** |  |
| **1,2,6, 9, 11, 12,13,2** | **carta=V=A, suma=11, suma=F=12** |  |
| **1,2,6,10, 11, 12,13,2** | **carta=V=A, suma=4, suma=15** |  |
| **C4: 1,2,3,8, 11,12,13,2** | **carta=F=J, suma=0, suma=10** |  |
| **C5: 1,2,3,4,8, 11,12,13,2** | **carta=Q, suma=0, suma=10** |  |
| **C6: 1,2,3,4,5, 8, 11,12,13,2** | **carta=K, suma=0, suma=10** |  |
| **C7: 1,2,3,4,5,7,11,12, 14, 15, F** | **carta=5, suma=16, suma=21** | **No es posible** |
| **C8: 1,2,3,4,5,7,11,12, 14, 16, F** |  |  |
| **C9: 1,2,3,4,5,7,11,12, 13, 14, 15 F** |  |  |

**5. Desarrolla los Casos de Prueba del Caso. El programa entregado prográmelo en Python**

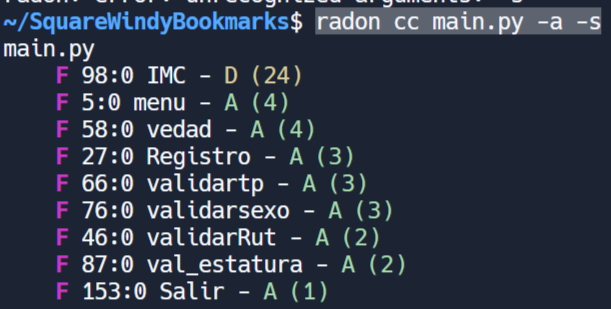
Conteste sobre la columna “salida” de la tabla de casos de prueba

**6. Usando Replit:** [**https://replit.com/**](https://replit.com/)**, instale desde la Shell la librería radón**

**Instalación:** $ pip install radon

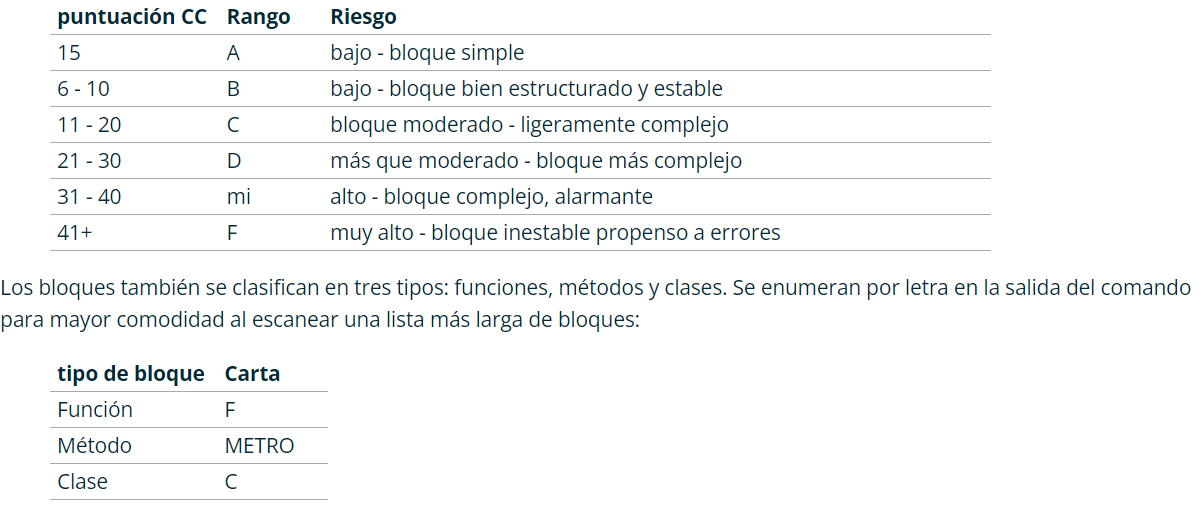
Ejecución de radon: radon cc nombreArchivo.py -a –s

Este es un ejemplo:



¿Cuál es la complejidad ciclomática (cc) de su propuesta?

Considere que la documentación de radón clasifica la complejidad de la siguiente forma:



**ANEXOS:**

Código algorítmico 21 real

Proceso Algoritmo21Real

Definir carta como caracter;

Definir valor\_carta como entero;

definir suma como entero;

suma<-0;

Repetir

Escribir "Ingrese carta como:A, [2,10], J, Q,K" ;

Leer carta;

Si carta = "A" entonces

si suma<10 entonces

valor\_carta<-11;

Sino

valor\_carta<-1;

FinSi

Sino Si carta="J" o carta="Q" o carta="K"

valor\_carta=10;

Sino

valor\_carta<-transformar(carta);

FinSi

FinSi

Suma<- valor\_carta+ suma;

Hasta Que suma = 21 o suma >21

Si suma=21 entonces

Escribir "Sus cartas suman ", suma, " puntos BLACKJACK - GANASTE";

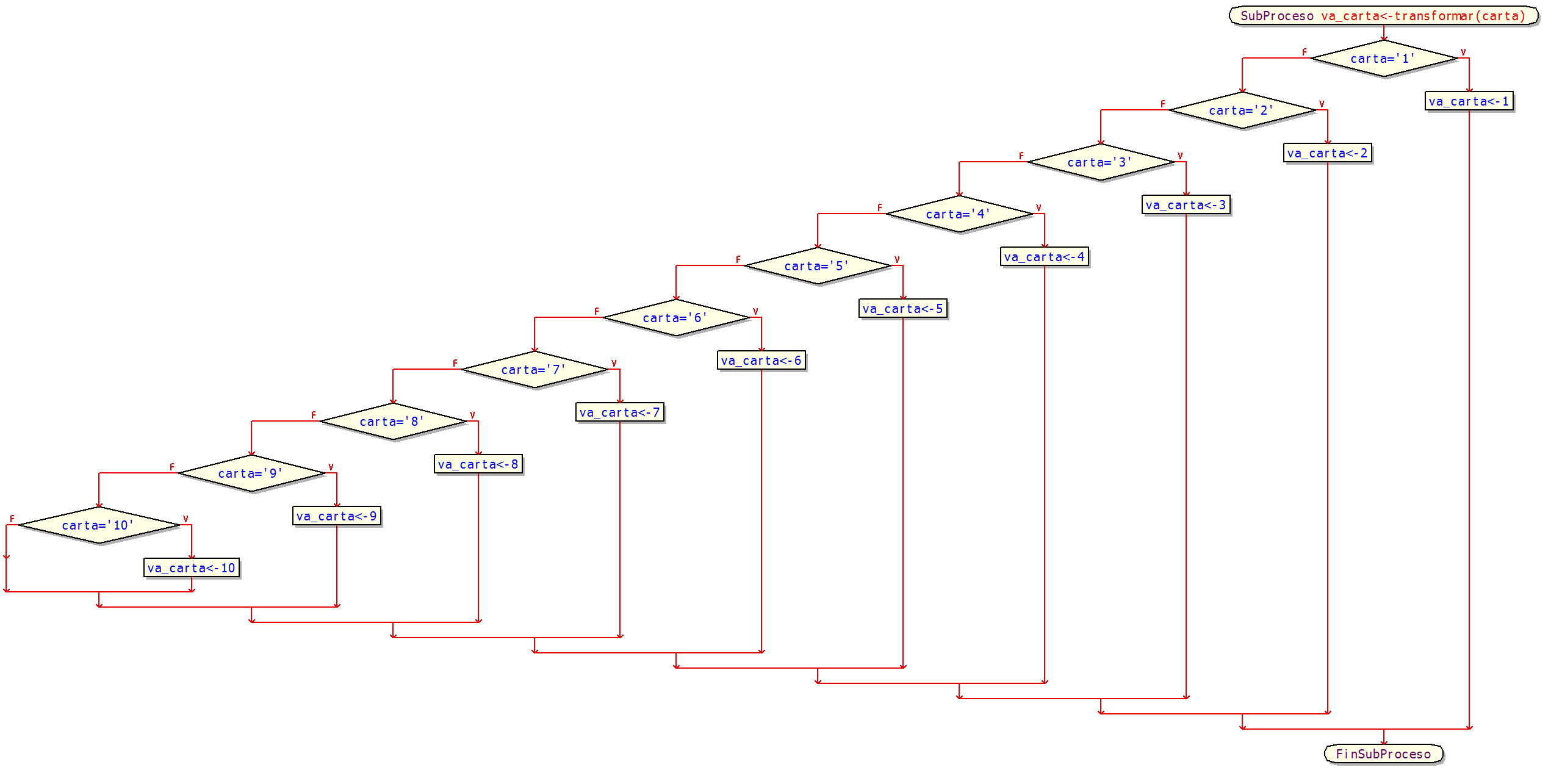
Sino

Escribir "Sus cartas suman ", suma, " puntos HAS PERDIDO";

FinSi

FinProceso

**Algoritmo función transformar**



SubProceso va\_carta<-transformar(carta)

si carta = "1" Entonces

va\_carta=1;

sino si carta="2" entonces

va\_carta=2;

Sino si carta="3" entonces

va\_carta=3;

Sino Si carta="4" entonces

va\_carta=4;

Sino Si carta="5" entonces

va\_carta=5;

Sino Si carta="6" entonces

va\_carta=6;

Sino Si carta="7" entonces

va\_carta=7;

Sino Si carta="8" entonces

va\_carta=8;

Sino Si carta="9" entonces

va\_carta=9;

Sino Si carta="10" entonces

va\_carta=10;

FinSi

FinSi

FinSi

FinSi

FinSi

FinSi

FinSi

FinSi

FinSi

FinSi

FinSubProceso