

Exercice 23 :

Soit la suite (u_n) définie par $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

Soit la suite (v_n) définie par $v_0 = 2$ $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

1°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (u_n) .

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n .

3°) Déterminez l'algorithme permettant de déterminer à partir de quel rang n les termes u_n de la suite atteignent ou dépassent n'importe quel nombre W .

4°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (v_n) . Que remarquez-vous ? Comment s'appelle ce type de suite ?

5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n .

Exo 23 : 1°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (u_n) .

$$u_0 = 0 \text{ et } u_{n+1} = 2u_n + 1$$

Menu \rightarrow RECUR \rightarrow $a_{n+1} = 2 \times a_n + 1$ EXE

\rightarrow SET \rightarrow Start 0 End 9 a_0 0 EXIT \rightarrow TABL

n	0	1	2	3	4	5	6	7	8	9
u_n										

Autre possibilité : 0 EXE 2 ANS + 1 EXE EXE EXE etc...

Exo 23 : 1°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (u_n) .

$$u_0 = 0 \text{ et } u_{n+1} = 2u_n + 1$$

Menu \rightarrow RECUR \rightarrow $a_{n+1} = 2 \times a_n + 1$ EXE

\rightarrow SET \rightarrow Start 0 End 9 a_0 0 EXIT \rightarrow TABL

n	0	1	2	3	4	5	6	7	8	9
u_n	0	1	3	7	15	31	63	127	255	511

Autre possibilité : 0 EXE 2 ANS + 1 EXE EXE EXE etc...

1°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (u_n) .

$$u_0 = 0 \text{ et } u_{n+1} = 2 u_n + 1$$

Menu → RECUR → $a_{n+1} = 2 \times a_n + 1$ EXE

→ SET → Start 0 End 9 a_0 0 EXIT → TABL

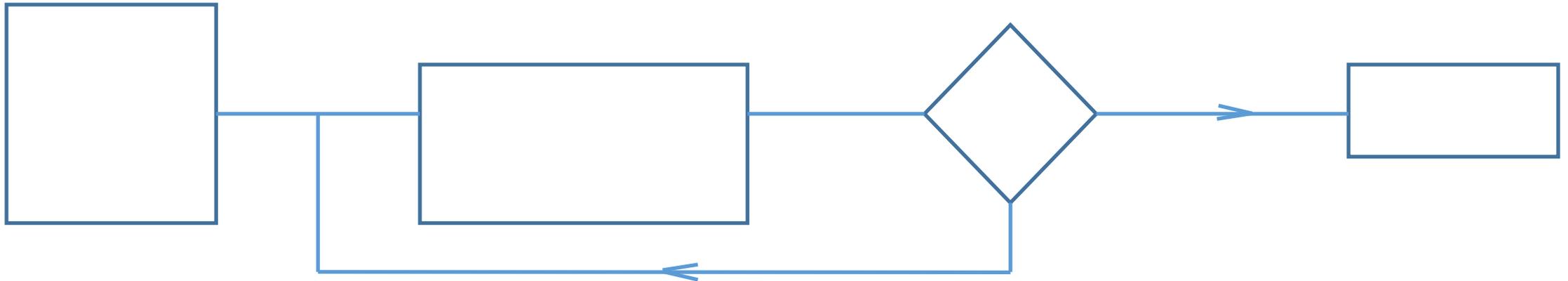
n	0	1	2	3	4	5	6	7	8	9
u_n	0	1	3	7	15	31	63	127	255	511

Pour afficher la courbe, on appuie ensuite sur G-CON (points de la suite reliés par la courbe de la fonction définie sur \mathbb{R}^+)

ou sur G-PLT (points non reliés).

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme



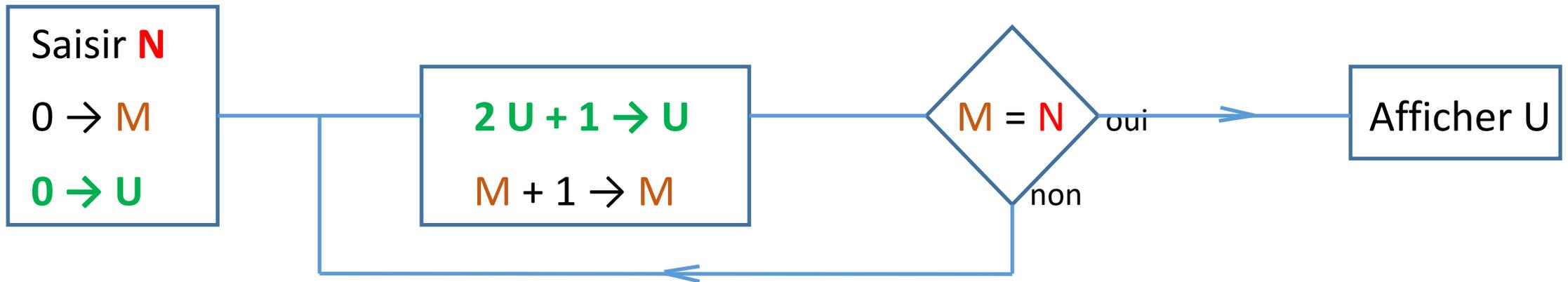
N : mémoire pour n

U : mémoire pour u_n

M : compteur de boucle

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme



N : mémoire pour n

U : mémoire pour u_n

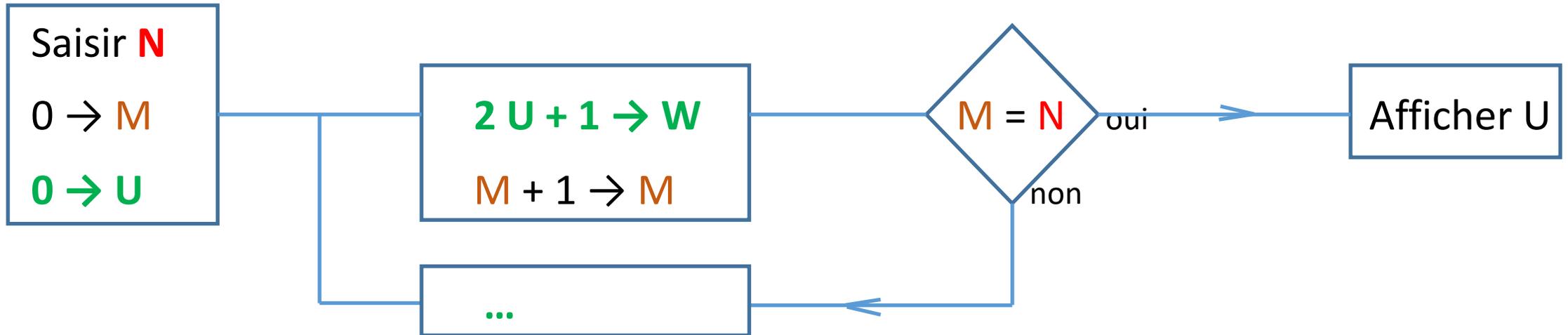
M : compteur de boucle

Remarque :

l'algorithme ne marche pas pour déterminer u_0 ! (la machine ne sort pas de la boucle alors que je sais déjà que $u_0 = 0$)

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme *Variante :*



N : mémoire pour n

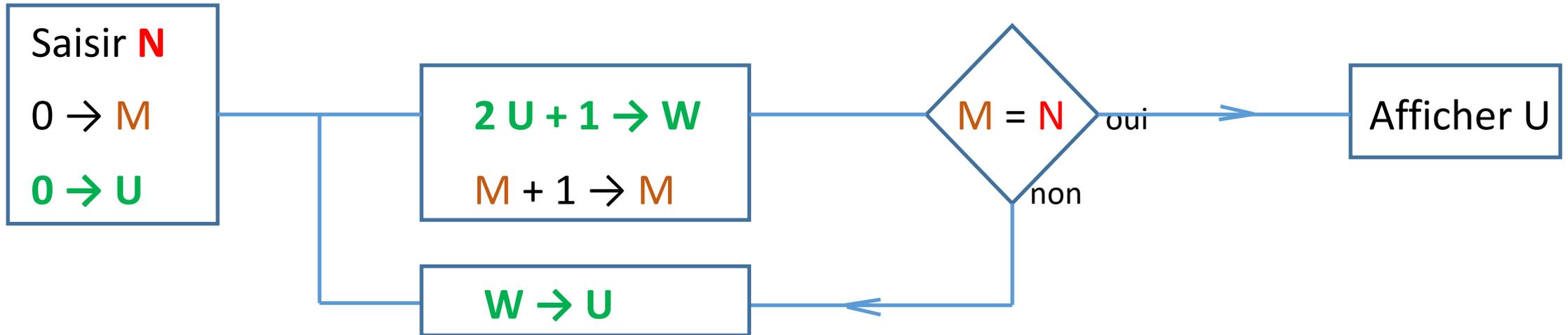
U : mémoire pour u_n

M : compteur de boucle

W : mémoire pour u_{n+1} qui est $2u_n + 1$

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme *Variante :*



N : mémoire pour n

U : mémoire pour u_n

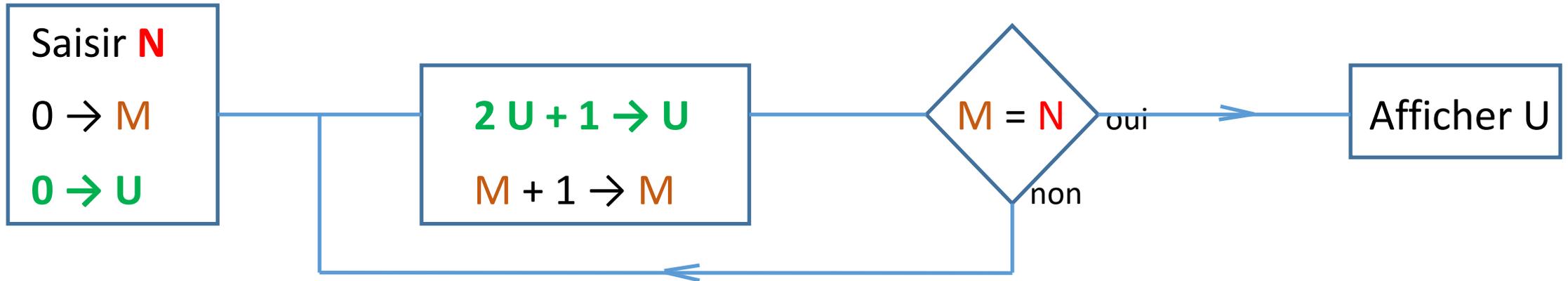
M : compteur de boucle

W : mémoire pour u_{n+1} qui est $2u_n + 1$

Pour le calcul suivant $u_{n+2} = 2u_{n+1} + 1$ u_{n+1} est le u_{n+1} de $u_{n+1} = 2u_n + 1$

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme



N : mémoire pour n

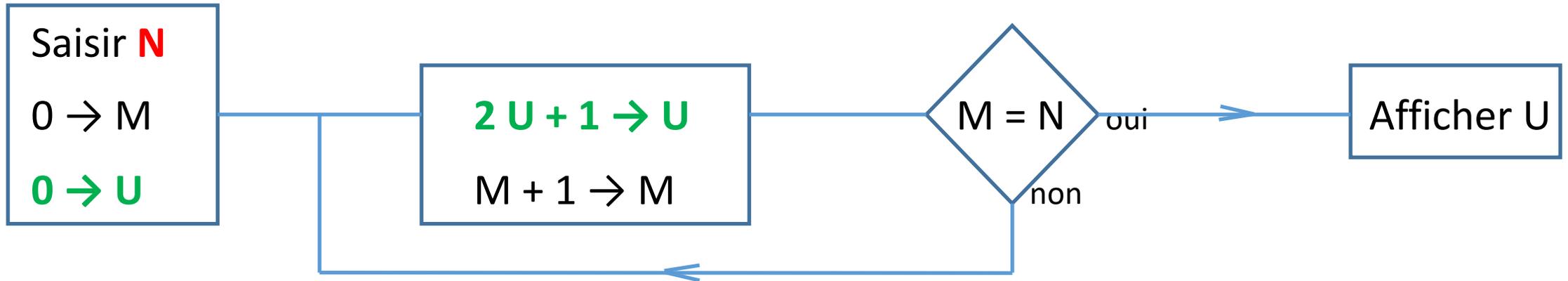
U : mémoire pour u_n

M : compteur de boucle

étape 2 : ... ?

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme



N : mémoire pour n

U : mémoire pour u_n

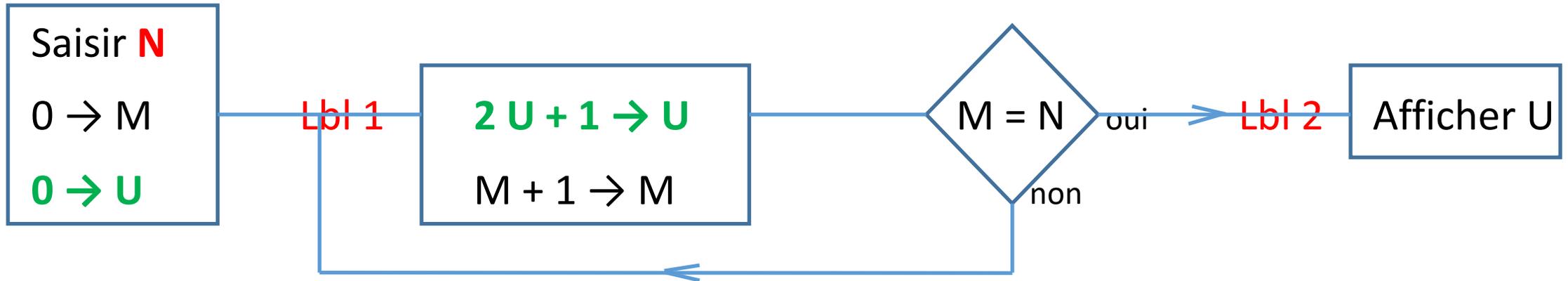
M : compteur de boucle

étape 2 : programme machine sur sa copie

...

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, U = u_n



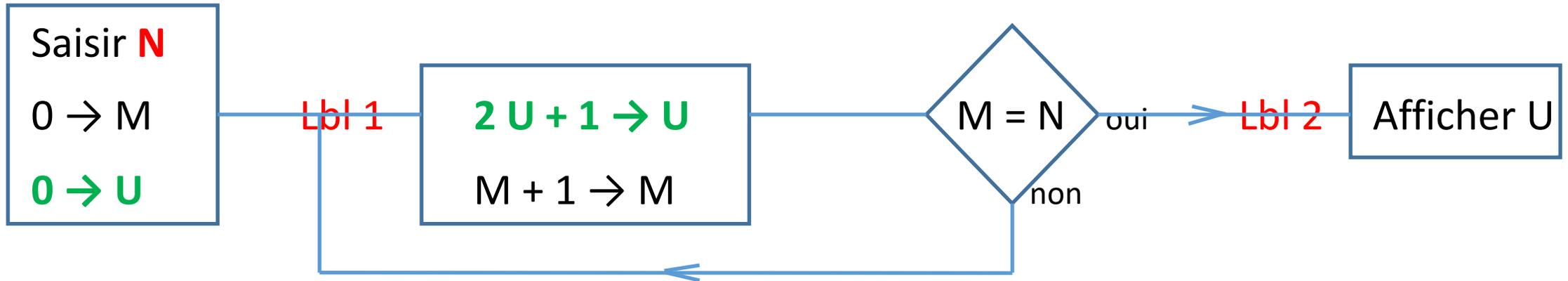
étape 2 : programme

? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, U = u_n



étape 2 : programme

? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape.

étape 1 : organigramme N est le n° n, M est le comptage de boucle, U = u_n



étape 2 : programme sur sa copie

? → N : 0 → M : 0 → U : Lbl 1 : 2 U + 1 → U : M + 1 → M :

If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape le programme dans sa machine

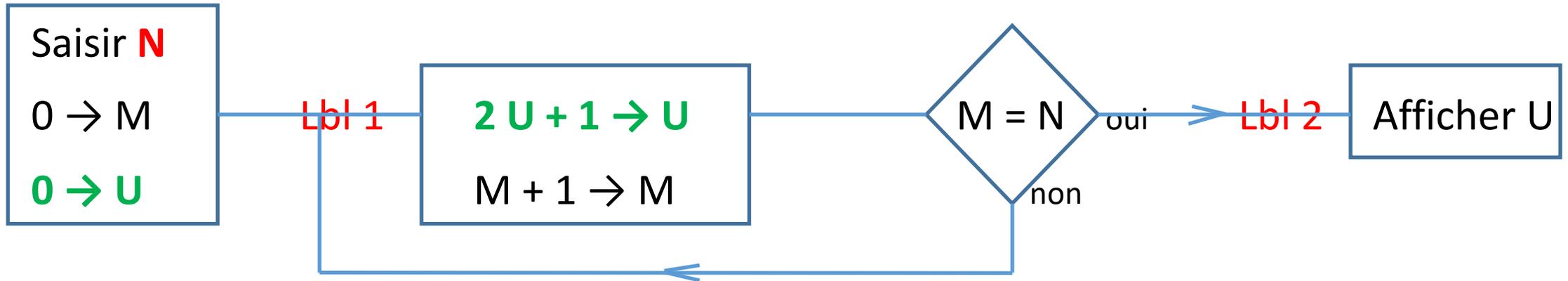
On allume la machine, dans le Menu on va dans PRGM, puis NEW, on tape le nom du programme puis EXE, puis on tape le programme.

Shift Prgm puis JUMP (Goto Lbl) ou COM (If Then Else)
ou REL (≥ :

Shift Prgm (? : ▲) → et " sont sur le clavier.

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, $U = u_n$



étape 2 : programme

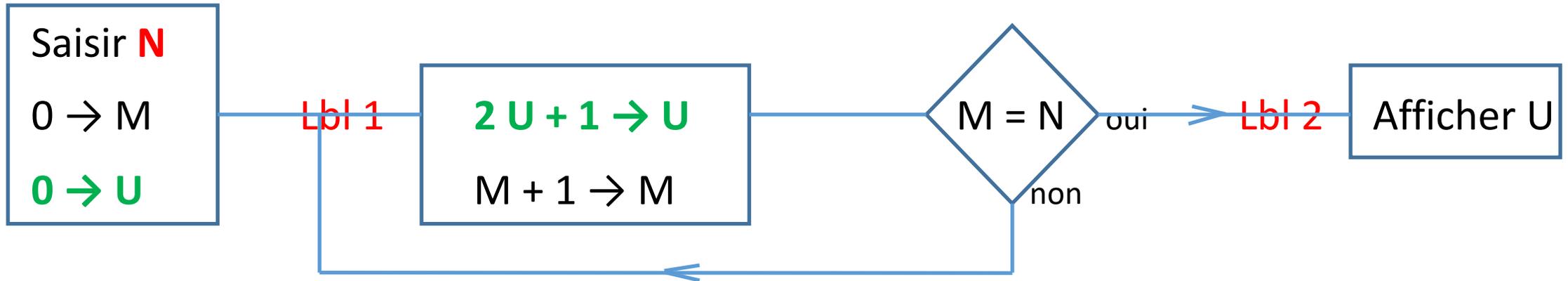
? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape. étape 4 : on teste.

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, $U = u_n$



étape 2 : programme

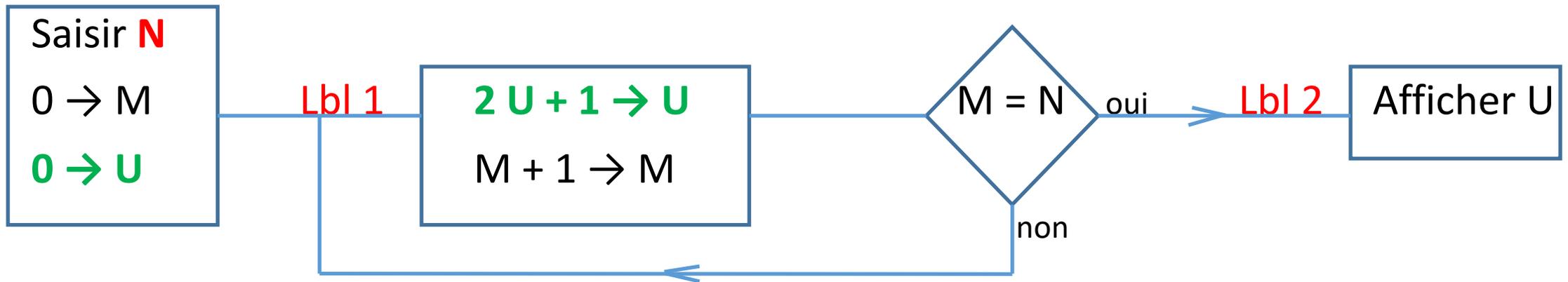
? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, U = u_n



étape 2 : programme

? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

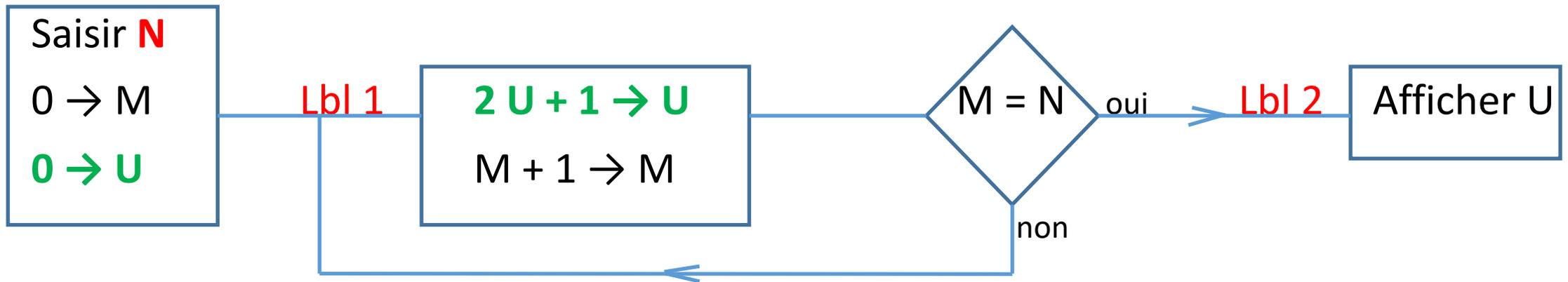
If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.

exemple $u_{30} = \dots$

2°) Déterminez l'algorithme permettant de déterminer n'importe quel terme u_n . $u_0 = 0$ et $u_{n+1} = 2u_n + 1$

étape 1 : organigramme N est le n° n, M est le comptage de boucle, U = u_n



étape 2 : programme

? → N : 0 → M : 0 → U : Lbl 1 : 2U + 1 → U : M + 1 → M :

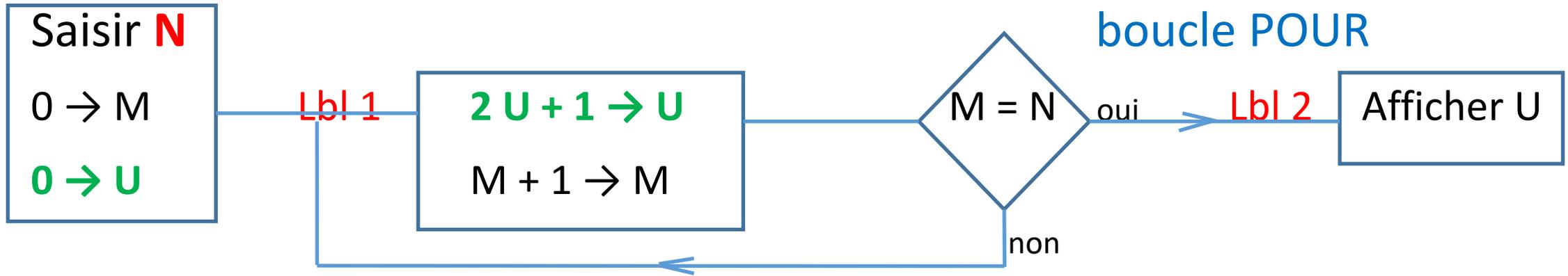
If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : U ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.

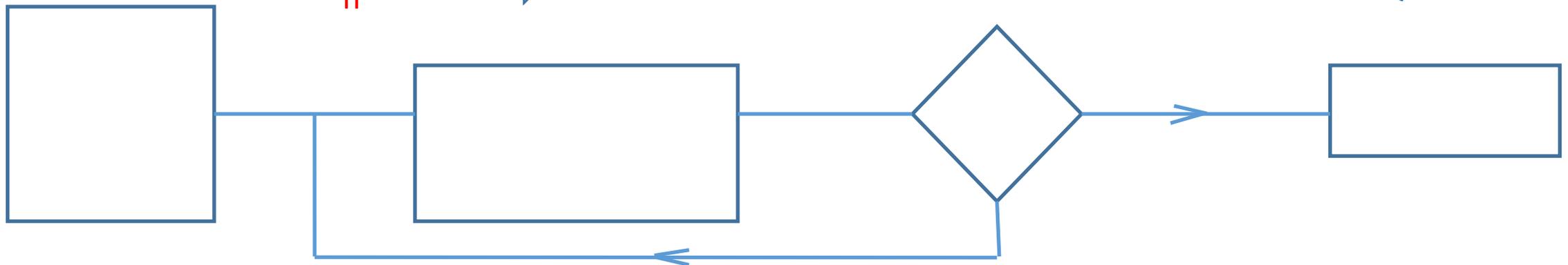
exemple $u_{30} = 1073741823$

3°) Déterminez l'algorithme permettant de déterminer à partir de quel rang n les termes u_n de la suite atteignent ou dépassent n'importe quel nombre W .

On modifie l'algorithme précédent : n fixé \Rightarrow on veut connaître u_n

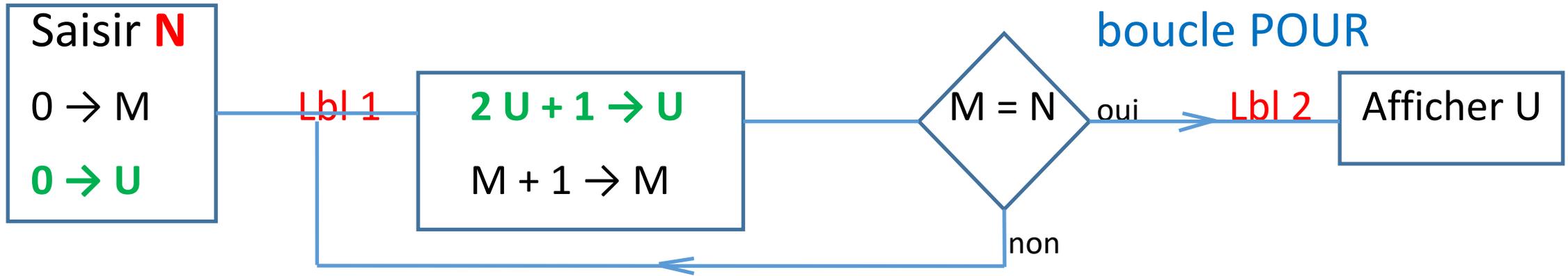


u_n fixé \Rightarrow on veut connaître n boucle TANT QUE

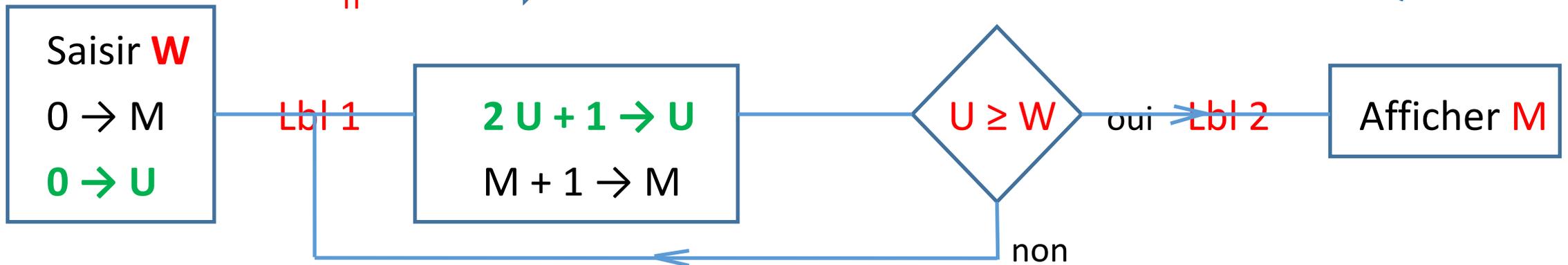


3°) Déterminez l'algorithme permettant de déterminer à partir de quel rang n les termes u_n de la suite atteignent ou dépassent n'importe quel nombre W .

On modifie l'algorithme précédent : n fixé \Rightarrow on veut connaître u_n

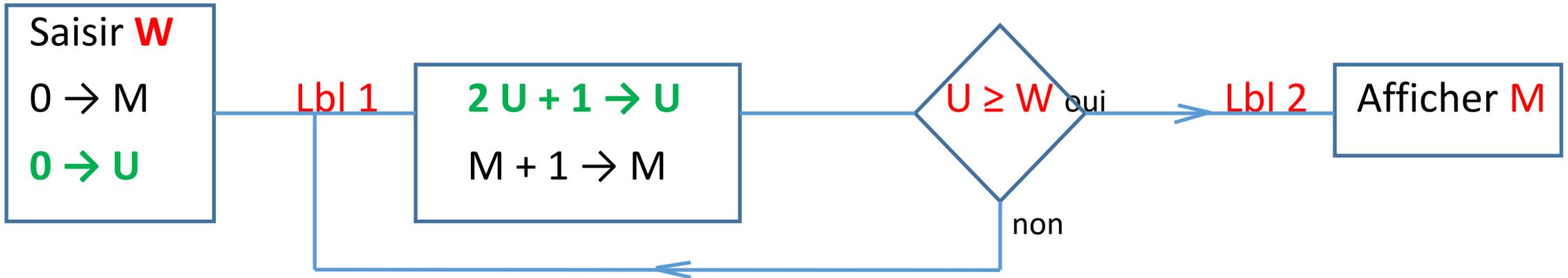


u_n fixé \Rightarrow on veut connaître n boucle TANT QUE



3°) Déterminez l'algorithme permettant de déterminer à partir de quel rang n les termes u_n de la suite atteignent ou dépassent n'importe quel nombre W .

étape 1 : organigramme $M =$ comptage de boucle, $U = u_n$, $W =$ le u_n voulu



étape 2 : programme

? → W : 0 → M : 0 → U : Lbl 1 : $2U + 1 \rightarrow U$: $M + 1 \rightarrow M$:

If $U \geq W$: Then Goto 2 : Else Goto 1 : Lbl 2 : M ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise

exemple $u_n \geq W = 10^9$ à partir de $n \geq M = 30$

4°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (v_n) .

$$v_0 = 2 ; v_1 = 4 \text{ et } v_{n+2} = v_{n+1} - v_n + 1$$

Menu → RECUR → Type → F3 ($a_{n+2} = Aa_{n+1} + Ba_n + \dots$) → F4
($n ; a_n ; a_{n+1} \dots$) F3 (a_{n+1}) - F2 (a_n) + 1 EXE
→ SET → Start 0 End 9 a_0 2 a_1 4 EXE → TABL

n	0	1	2	3	4	5	6	7	8	9
v_n	2	4	3	0	-2	-1	2	4	3	0

4°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (v_n) .

2 ; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

Menu → RECUR → Type → F3 ($a_{n+2} = Aa_{n+1} + Ba_n + \dots$) → F4 ($n ; a_n ; a_{n+1} \dots$) F3 (a_{n+1}) - F2 (a_n) + 1 EXE

→ SET → Start 0 End 9 a_0 2 a_1 4 EXE → TABL

n	0	1	2	3	4	5	6	7	8	9
v_n	2	4	3	0	-2	-1	2	4	3	0

Que remarquez-vous ?

4°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (v_n) .

$$v_0 = 2 ; v_1 = 4 \text{ et } v_{n+2} = v_{n+1} - v_n + 1$$

Menu \rightarrow RECUR \rightarrow Type \rightarrow F3 ($a_{n+2} = Aa_{n+1} + Ba_n + \dots$) \rightarrow F4
($n ; a_n ; a_{n+1} \dots$) F3 (a_{n+1}) $-$ F2 (a_n) $+ 1$ EXE \rightarrow SET \rightarrow
Start 0 End 9 a_0 2 a_1 4 EXE \rightarrow TABL

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
v_n	2	4	3	0	-2	-1	2	4	3	0	-2	-1	2	4

Que remarquez-vous ?

Les termes reviennent par séries de 6 :

4°) Déterminez avec le tableur de votre calculatrice les 10 premiers termes de la suite (v_n) .

$$v_0 = 2 ; v_1 = 4 \text{ et } v_{n+2} = v_{n+1} - v_n + 1$$

Que remarquez-vous ?

Les termes reviennent par séries de 6 :

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
v_n	2	4	3	0	-2	-1	2	4	3	0	-2	-1	2	4

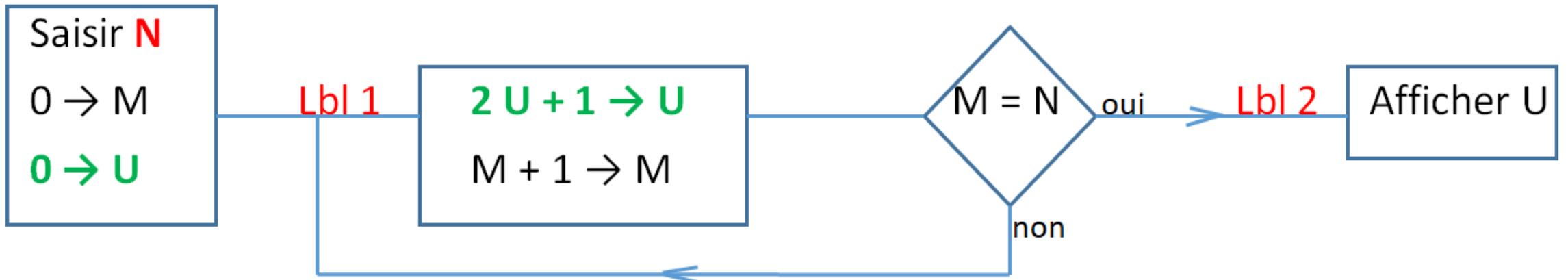
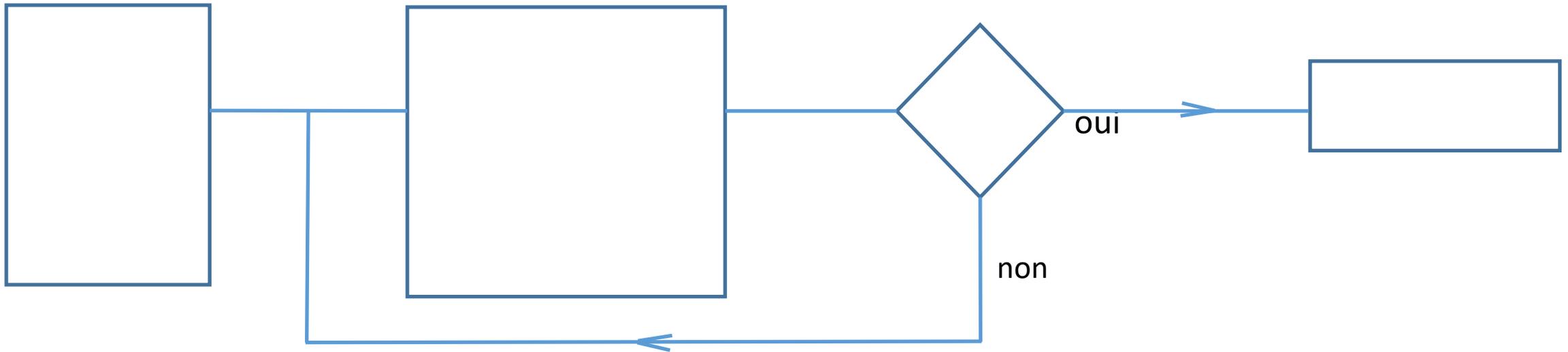
Comment s'appelle ce type de suite ?

Les suites **périodiques** (de période 6).

$$v_{n+6k} = v_n \quad \text{pour tout } n \text{ et tout entier positif } k.$$

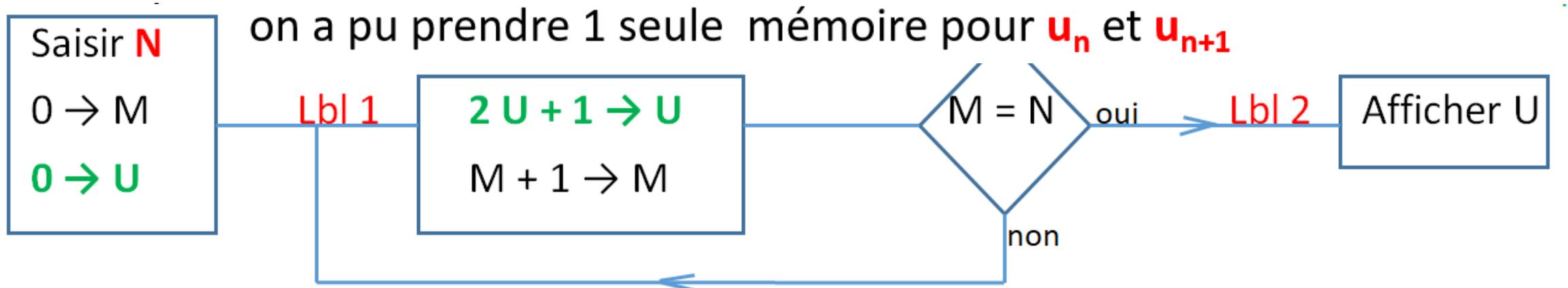
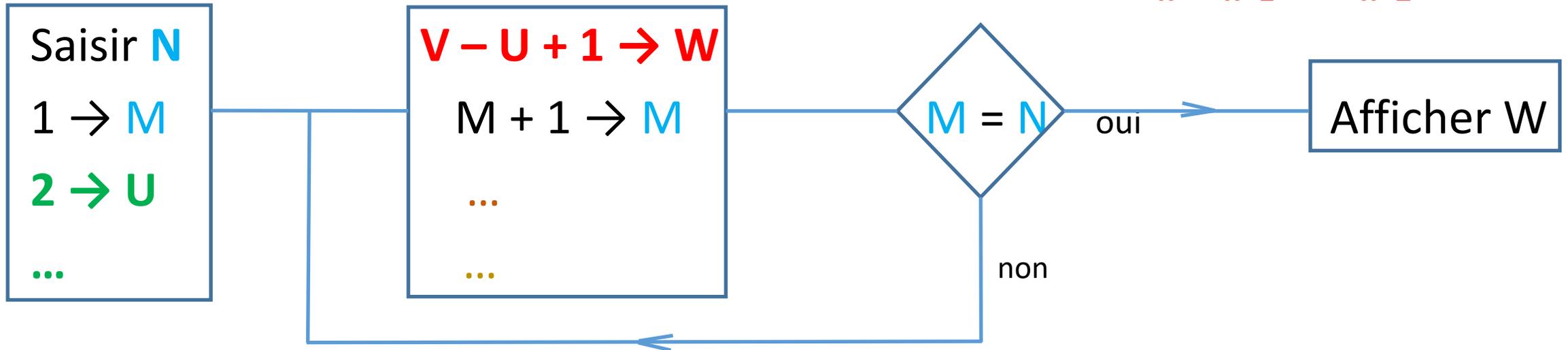
5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

étape 1 : organigramme (en s'inspirant de celui de la suite récurrente d'ordre 1)



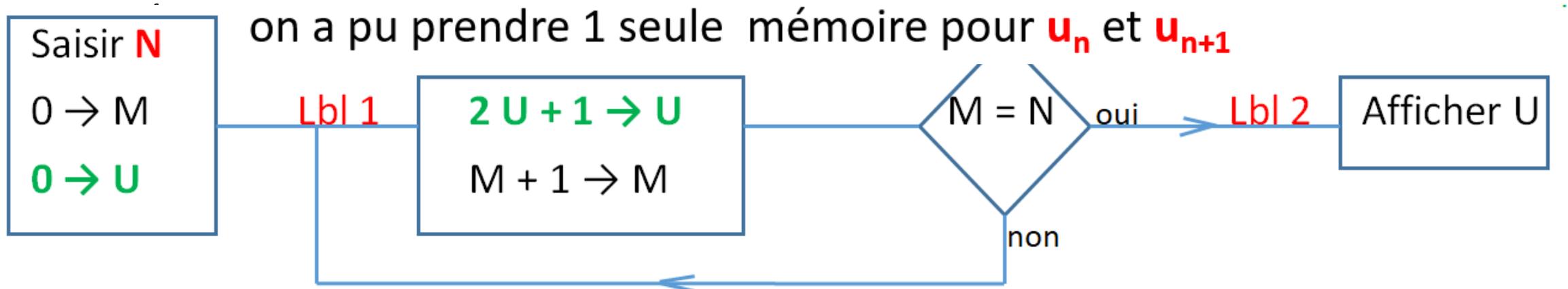
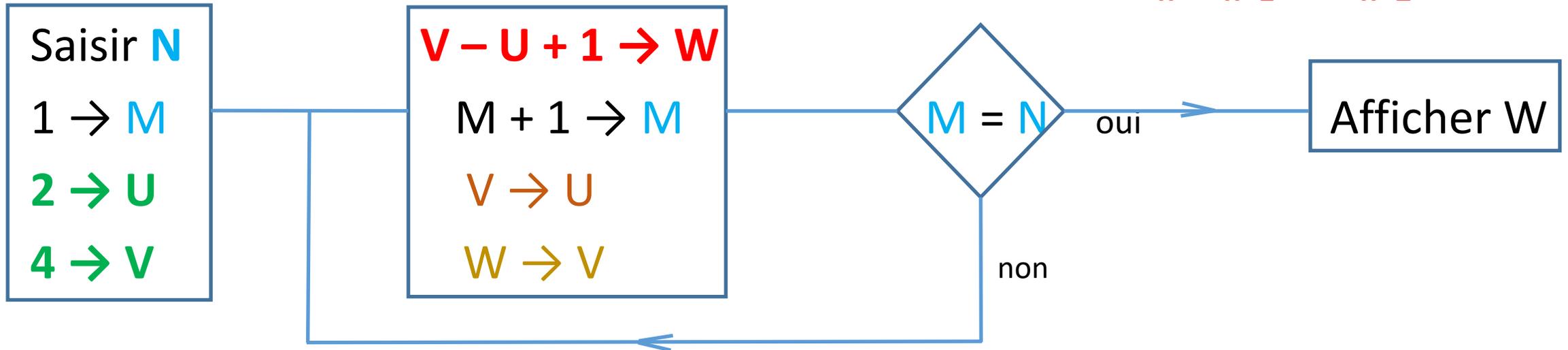
5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

étape 1 : on est obligé de prendre 3 mémoires \neq pour v_n , v_{n+1} et v_{n+2}

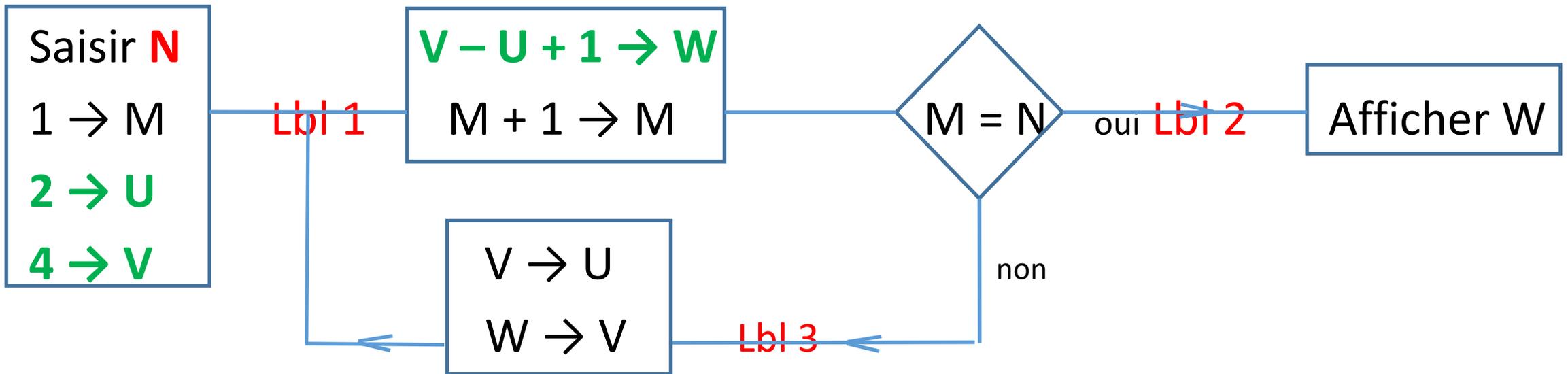


5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

étape 1 : on est obligé de prendre 3 mémoires \neq pour v_n , v_{n+1} et v_{n+2}

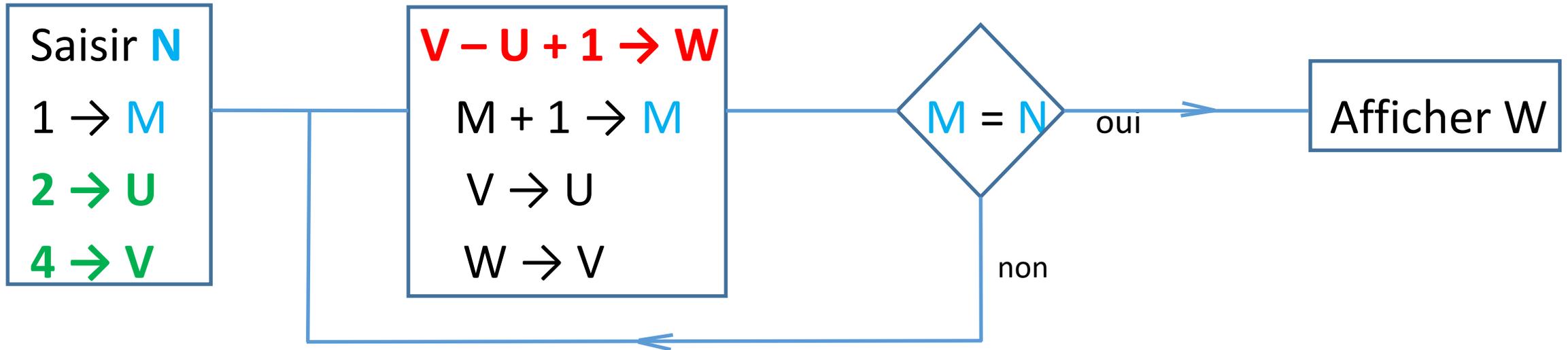


Variante :



5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

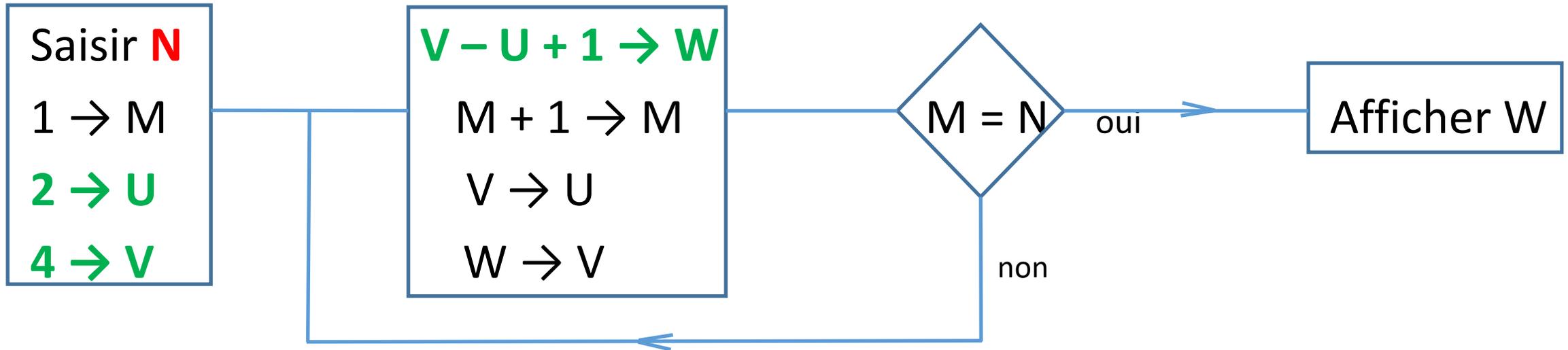
étape 1 : organigramme



Remarque : Peut-on intervertir certaines actions ?

5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

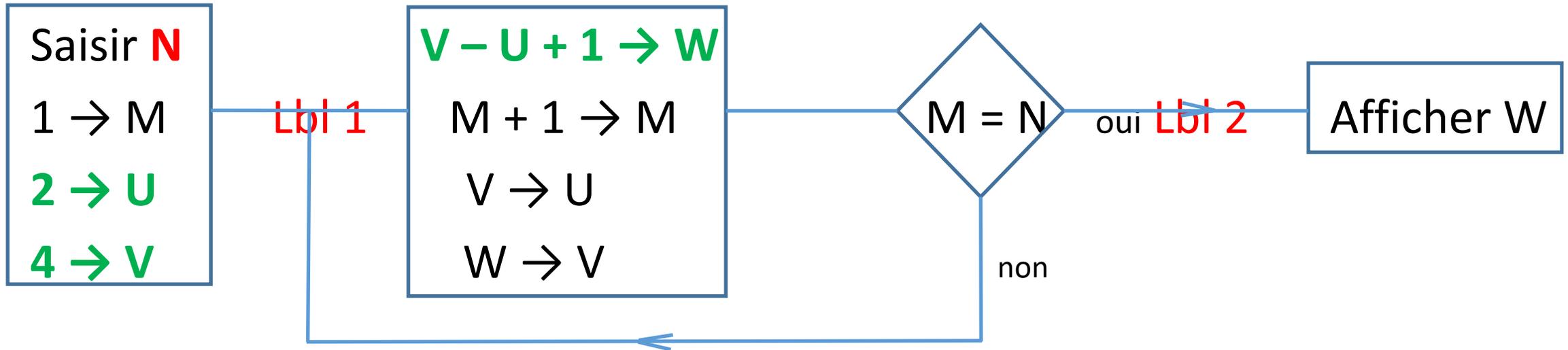
étape 1 : organigramme



Remarque : on ne peut intervertir les 4 actions de la boucle, car si l'on fait $W \rightarrow V$ puis $V \rightarrow U$, V a pris la valeur W , donc ensuite U aussi, donc on sera dans le cas faux $V = U = W$; idem pour $V - U + 1 \rightarrow W$ qu'il faut faire avec les valeurs U et V **avant** de déterminer les U et V de l'**étape suivante**.

5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n . $v_0 = 2$; $v_1 = 4$ et $v_{n+2} = v_{n+1} - v_n + 1$

étape 1 : organigramme



étape 2 : programme

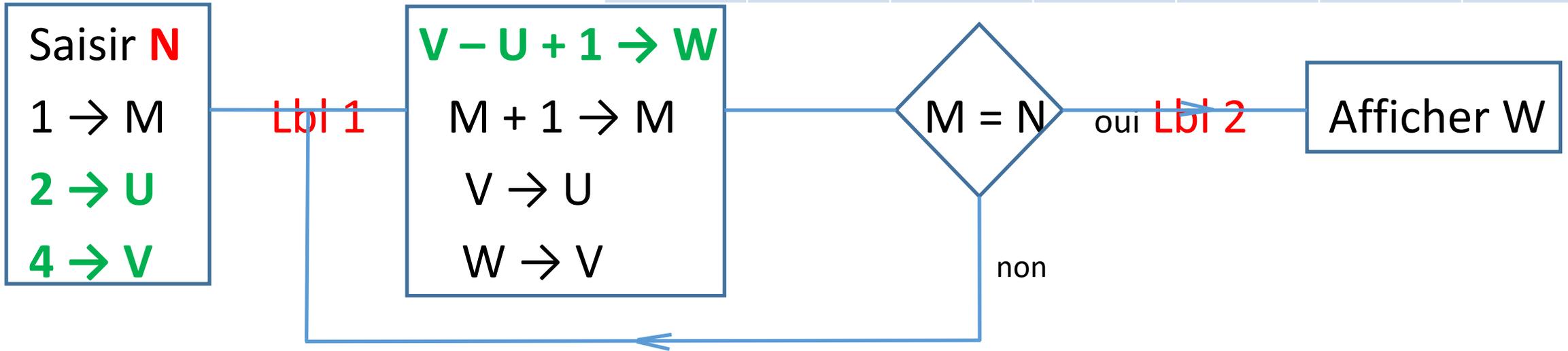
? → N : 1 → M : 2 → U : 4 → V : Lbl 1 : V - U + 1 → W : M + 1 → M : V → U :
W → V : If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : W ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.

5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n .

n	2	5	18	23	159	200
v_n						

étape 1 : organigramme



étape 2 : programme

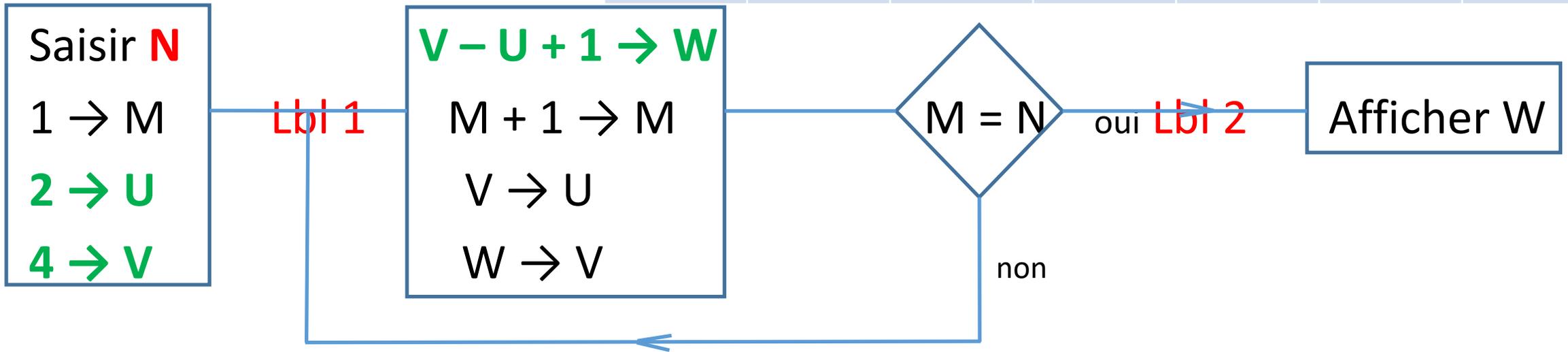
? → N : 1 → M : 2 → U : 4 → V : Lbl 1 : $V - U + 1 \rightarrow W$: $M + 1 \rightarrow M$: $V \rightarrow U$:
 $W \rightarrow V$: If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : W ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.

5°) Déterminez l'algorithme permettant de déterminer n'importe quel terme v_n .

n	2	5	18	23	159	200
v_n	3	-1	2	-1	0	3

étape 1 : organigramme



étape 2 : programme

? → N : 1 → M : 2 → U : 4 → V : Lbl 1 : V - U + 1 → W : M + 1 → M : V → U :
W → V : If M = N : Then Goto 2 : Else Goto 1 : Lbl 2 : W ▲

étape 3 : on tape. étape 4 : on teste. étape 5 : on utilise.