

Programme utilisateur n°1.

A lgorithme prévu comme démonstrateur d'aiguillage, cette feuille perforée est prévue pour héberger deux logiciels binaires élémentaires. Sur la Fig.1 un premier pion de donnée en jaune sert à sélectionner le cas traité par le programme. Puis on configure une ou

Pion de sélection

Fig.1

plusieurs séparations de la donnée, ici représentée en rouge, par un ou des pions à l'état "B". La donnée BINAIRE est quelconque et comporte un nombre libre de BITS.

➤ Utilisation.

P lacer le BIT de sélection à l'état "0" ou à l'état "1". Grand classique de base en traitements binaires, le comportement de l'algorithme est le suivant :

- Si le bit est à "0" l'intégralité des BITS de la donnée est forcée à l'état "0".
- Si le bit est à "1" l'intégralité des BITS de la donnée est configurée à l'état "1".
- Le processus se déroulera jusqu'à ce qu'une LECTURE détecte un état séparateur "B".

Interprétation des diagrammes.

G lobalement les sites Internet qui traitent de la machine de Turing utilisent presque tous une *représentation du comportement* des algorithmes proposés *qui est le suivant* :

- Bien que ce ne soit pas un impératif absolu, par convention tout programme devrait débuter par la transition n°1 et le point d'entrée est représenté par la flèche facultative ➡.
- Les différents états de TRANSITION sont numérotés et symbolisés par : **N**
- Les TRANSITIONS sont précisées par des flèches curvilignes. Si la flèche recircule sur le même état, alors il n'y a pas de transition.

➤ Symbolisation des actions.

Elles sont codées globalement sous la forme :

|| Si Lecture = **X**: faire **E**, **D**.
X: Lecture de l'un des états B, 0 ou 1.
E pour écrire un "B", un "0" ou un "1".
D pour déplacer à **G**auche ou à **D**roite.
F : Déclare la **F**in du programme."

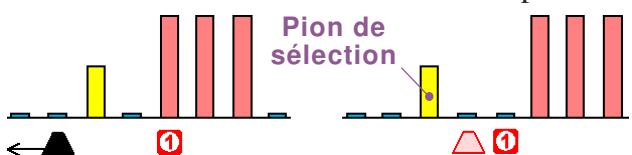
Exemples : **B: G** (Une seule action.)
B: 0,D (Deux actions.)

Programme utilisateur n°2.

D éplacer une suite de "1" d'une case à Droite si un pion de sélection est à "0" et *mouvoir la tête de L/E à la fin d'une deuxième donnée* si le BIT d'option est à "1".

➤ Utilisation pour l'option "0".

Le BIT de sélection est placé à "0". *La donnée comporte un nombre quelconque de "1" mais attention, déjà avec 3 BITS les mouvements de la tête de L/E sont importants.*



Fondamentalement ce programme réalise le SHIFT à droite d'une suite de "1".

➤ Utilisation pour l'option "1".

Le BIT de sélection est placé à "1". *Deux données BINAIREs suivent le BIT d'option. Ce programme déplace la tête L/E à la fin de la deuxième donnée.*

Fondamentalement cet algorithme simple correspond au pointage du poids faible d'une donnée BINAIRE parmi plusieurs.

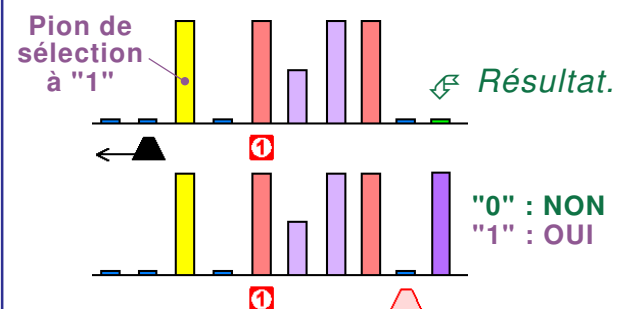
Programme utilisateur n°3.

A lgorithme avec aiguillage pour deux options possibles. Banale dans son comportement, le traitement pour "0" consiste à inverser tous les BITS d'une chaîne Binaire.



➤ Utilisation pour l'option "1".

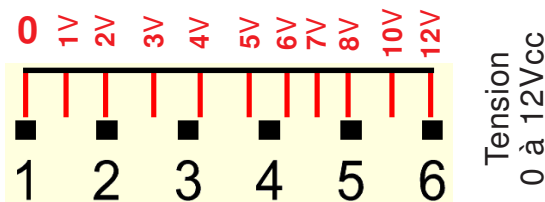
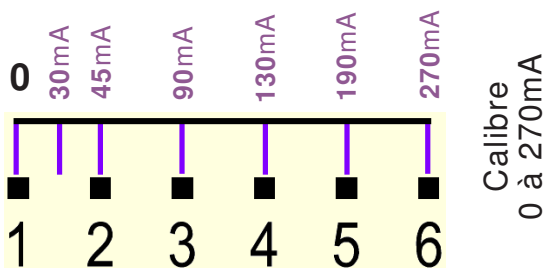
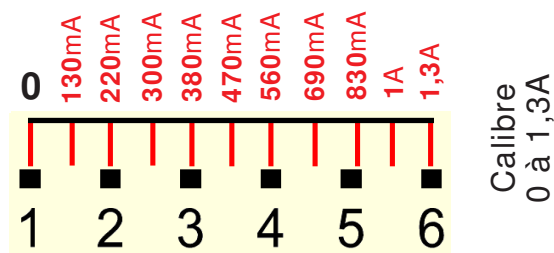
Le BIT de sélection étant placé à "1" ce programme détecte si une sous-chaîne [01] est présente dans une donnée Binaire.



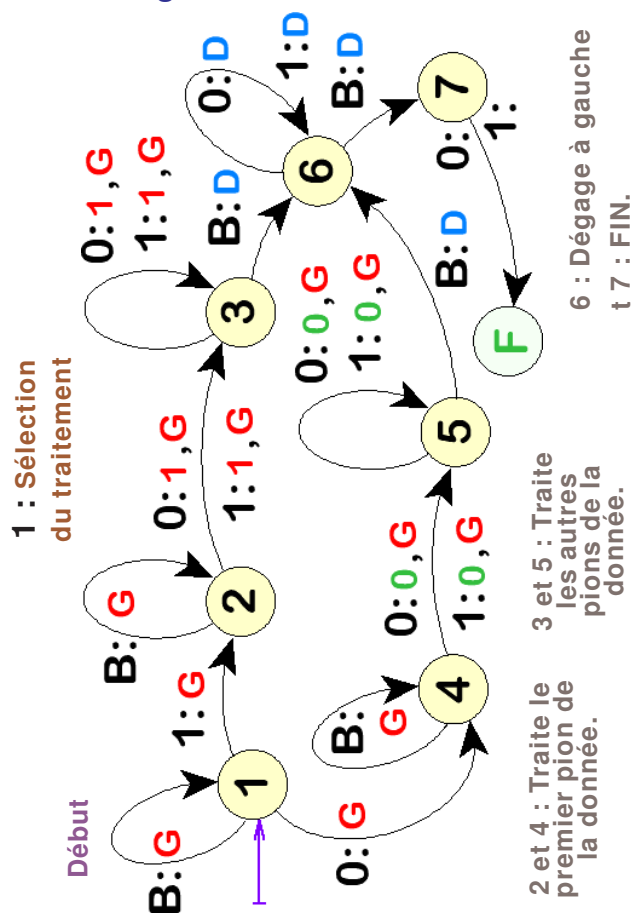
Quelle que soit la position éventuelle dans la donnée, le résultat sera écrit à droite de la valeur Binaire sous forme d'un BIT booléen.

Mesurages électriques.

➤ Calibration du galvanomètre.



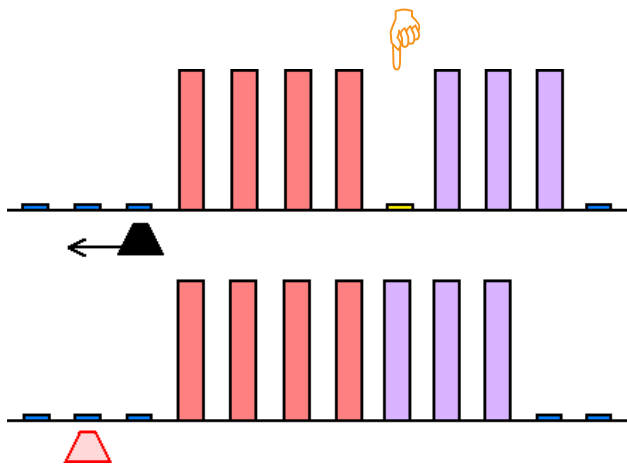
Programme utilisateur n°1.



Programme utilisateur n°4.

Bien que ressemblant à une addition de deux Entiers Unaires, ce programme effectue fondamentalement la **concaténation de deux chaînes** composées exclusivement de BITS positionnés "1".

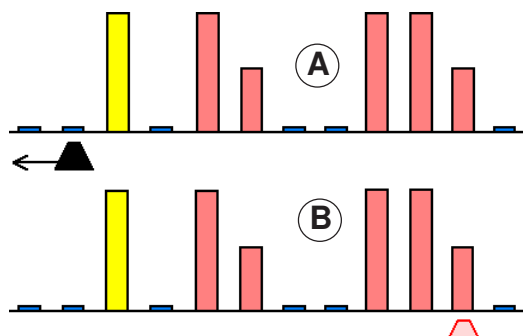
ATTENTION : Le séparateur entre les deux chaînes Unaires ne doit comporter qu'un seul BIT positionné à "B".



De façon banale, la tête Lecture/Écriture est positionnée à gauche des données pour faire sortir à droite le résultat.

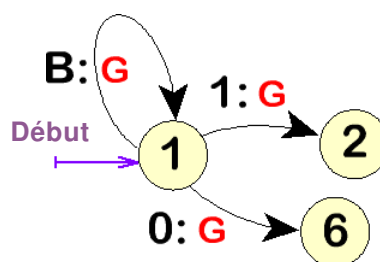
Programme utilisateur n°2.

Ci-dessous en **A** la configuration de départ et en **B** celle d'achèvement du traitement qui consiste à pointer la fin de la deuxième donnée BINAIRE.



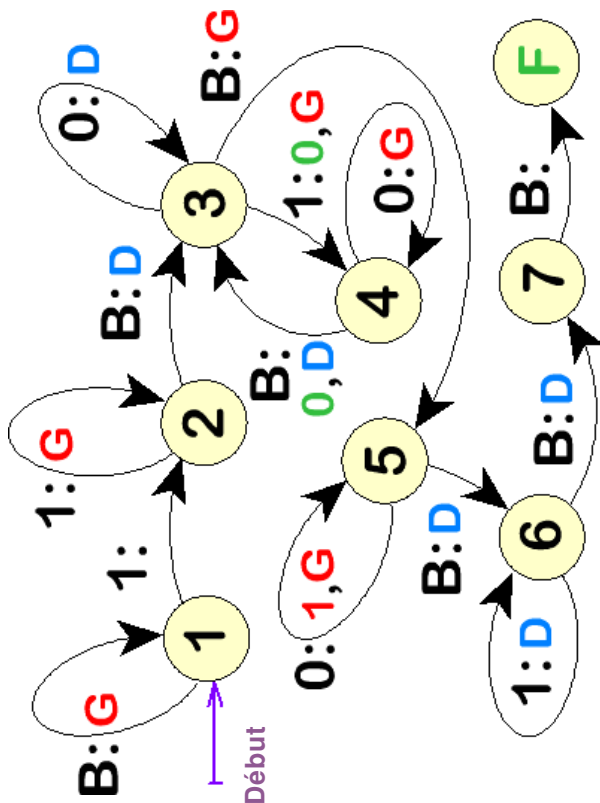
➤ Aiguillage de l'option.

Ce traitement laisse la tête de L/E sur le premier séparateur qui suit le BIT d'option.



Programme utilisateur n°6.

Remplacer la valeur d'un Entier codé sous forme Unaire par son double.



Programme utilisateur n°7.

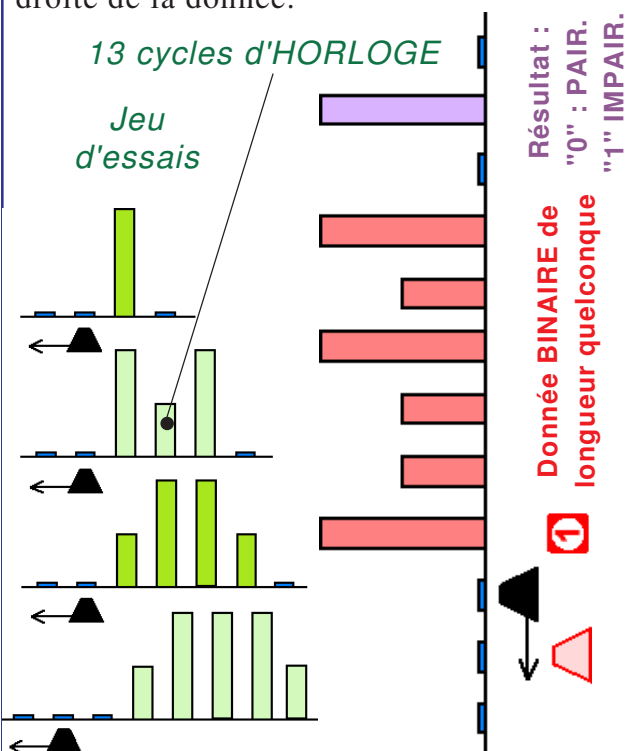
N	Binaire pur	Code GRAI
00	0000	0000
01	0001	0001
02	0010	0011
03	0011	0010
04	0100	0110
05	0101	0111
06	0110	0101
07	0111	0100
08	1000	1100
09	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Programme utilisateur n°5.

Détermine la PARITÉ du nombre de "1" dans une chaîne BINAIRE de longueur quelconque. Prévoir au moins trois "B" à droite de la donnée.

13 cycles d'HORLOGE

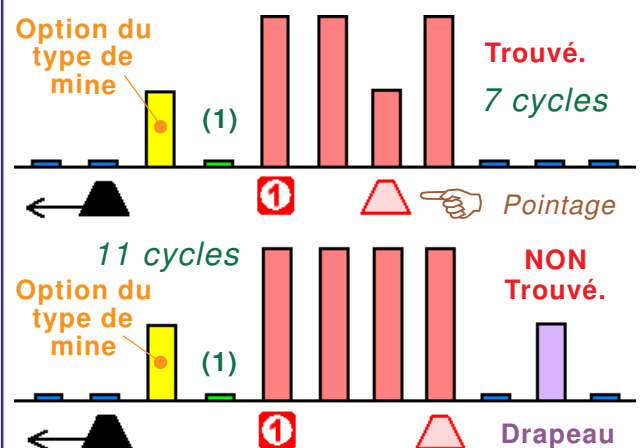
Jeu d'essais



Programme utilisateur n°8.

Programme de Déminage. Un premier BIT d'aiguillage permet de choisir l'option du type de mines. Avec "0" recherche les mines de type "0", et avec "1" détecte les mines de type "1". Si aucune mine n'est trouvée dans le champ, positionne un Drapeau dont le type est celui de l'option de recherche. Si une mine est trouvée, la tête de Lecture/Écriture pointe son emplacement.

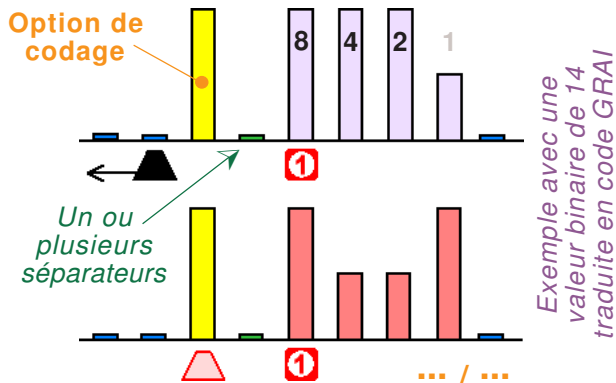
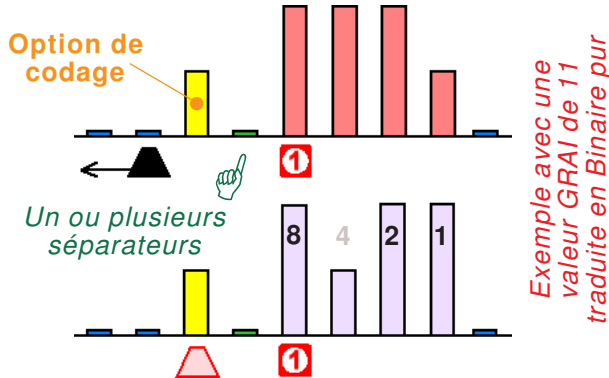
Exemples pour "0" :



(1) : Nombre quelconque de séparateurs "B" entre le BIT d'option et le champ de mines.

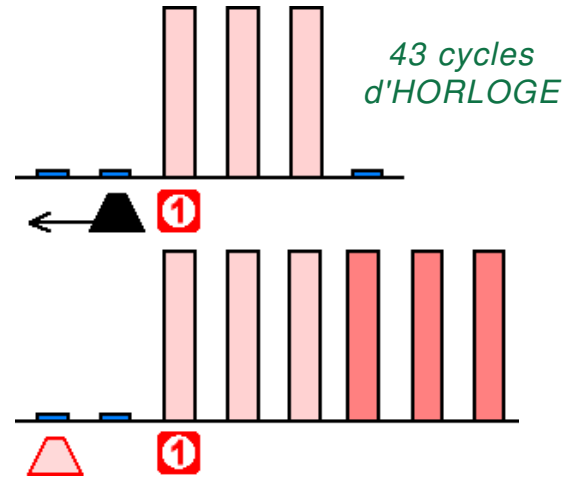
Programme utilisateur n°7.

Logiciel avec aiguillage qui *pour "0" translate un code GRAI vers du Binaire pur et qui pour "1" transforme un Binaire pur vers son équivalent en code GRAI.*



Programme utilisateur n°6.

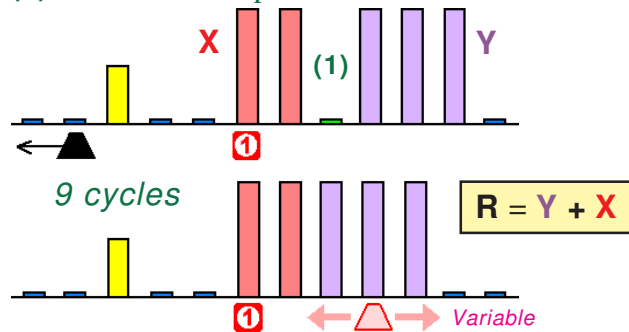
Algorithme agencé pour doubler une chaîne binaire constituée d'une suite de "1". La tête termine deux séparateurs à gauche de la donnée. Éviter plus de quatre BITS à "1" pour éviter un nombre de mouvements prohibitif sur la machine. Par exemple pour un entier initial de 3 l'HORLOGE système réalise déjà 42 cycles d'instructions.



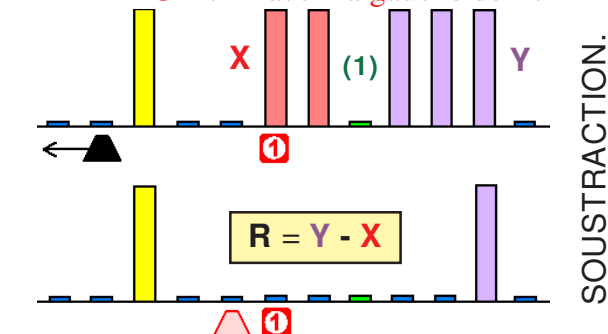
Fondamentalement ce programme réalise la *multiplication d'un Entier codé sous forme Unaire par deux.*

Programme utilisateur n°9.

Algorithme avec sélecteur qui *pour "0" effectue une ADDITION Unaire et qui pour "1" fait une SOUSTRACTION Unaire.*
(1) : UN SEUL séparateurs "B" entre X et Y.

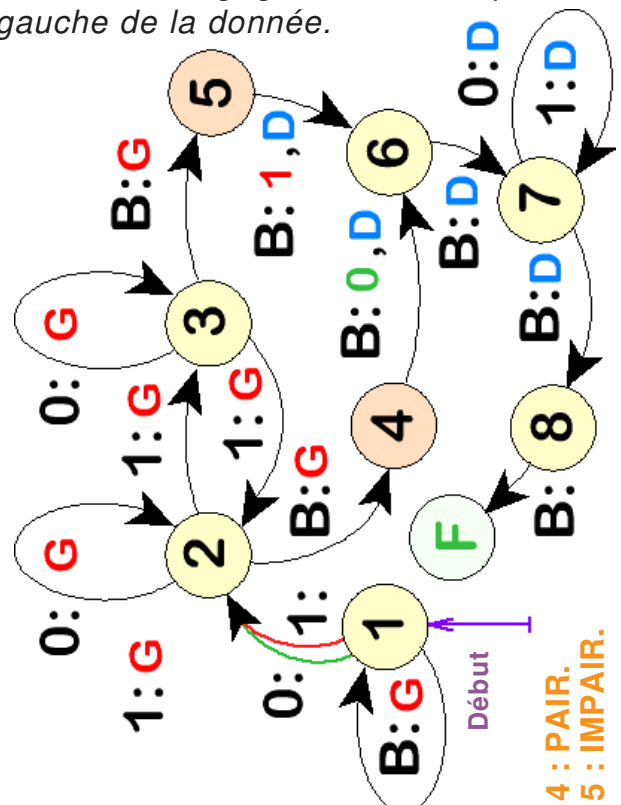


ATTENTION : Il faut X Inférieur ou égal à Y.
ATTENTION : Il faut X à gauche de Y.



Programme utilisateur n°5.

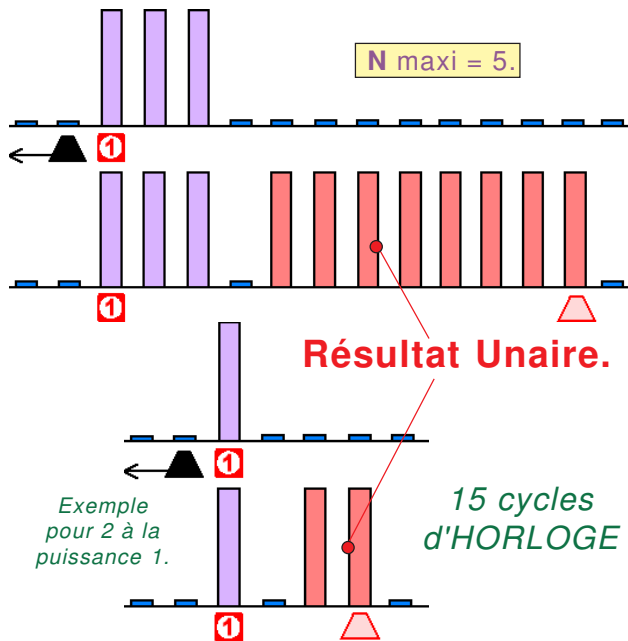
1 : Cherche le premier BIT.
6, 7 et 8 : Dégagent de deux pions à gauche de la donnée.



Programme utilisateur n°10.

Programme qui calcule la valeur de 2 élevé à la puissance N . La valeur de l'exposant N est exprimée en Unaire. Éviter de choisir N supérieur à 3 car cet algorithme génère beaucoup de mouvements machine.

ATTENTION : Il faut laisser un nombre de séparateurs "B" suffisant à droite de N .

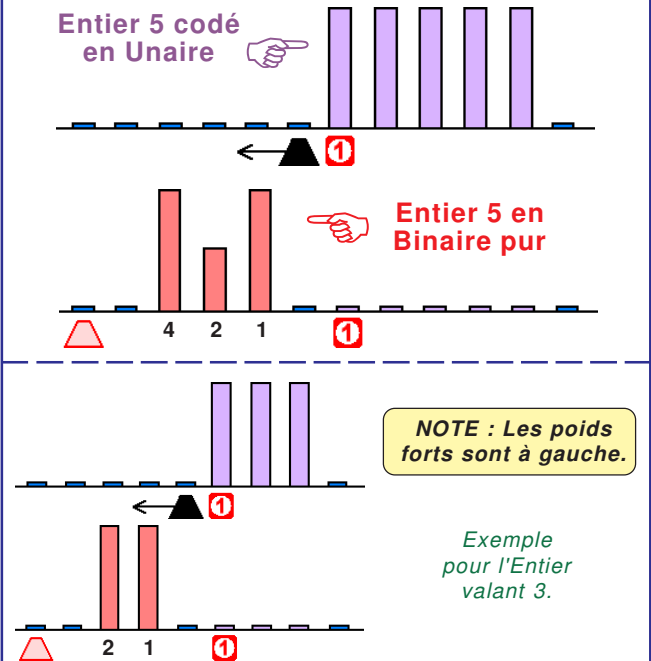


Programme utilisateur n°12.

Algorithme de transcodage en BINAIRE pur d'un Entier indiqué en Unaire.

(Efface la donnée initiale.) Prévoir de l'espace de type "B" à gauche de la donnée initiale en fonction du résultat attendu.

ATTENTION : Déjà 65 mouvements machine pour la valeur entière 5.

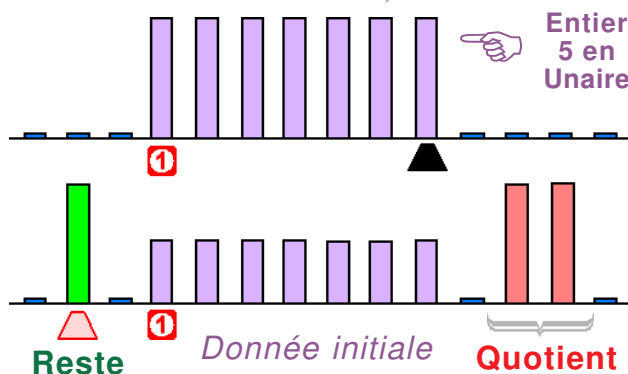


Programme utilisateur n°14.

Algorithme qui effectue la division euclidienne par 3 d'un Entier exprimé en Unaire. Le Quotient est positionné à droite de la donnée. Les BITS de la valeur initiale sont tous remplacés par des "0".

Il faut prévoir au moins trois espaces de type "B" à gauche de la donnée pour pouvoir "dégager" la valeur du Reste et $Q+2$ séparateurs de type "B" à droite de la donnée pour coder la valeur qui résulte de la division entière.

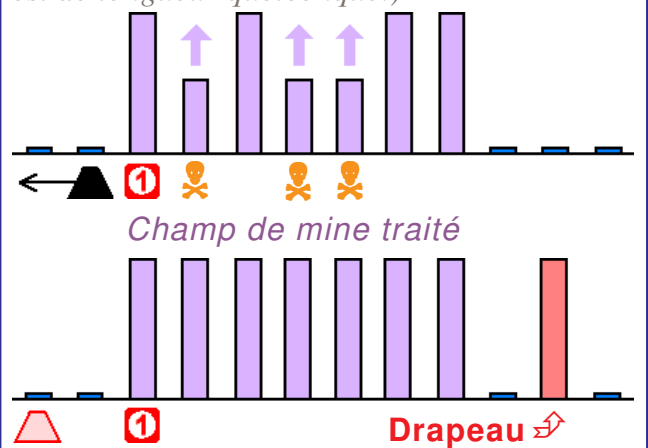
(Pour la valeur entière de 7 effectuée déjà 33 instructions sur la machine.)



ATTENTION : La tête de lecture/Ecriture doit se trouver à Droite de la Donnée.

Programme utilisateur n°17.

Petit programme de Déminage assez voisin du programme n°8. Toutefois il ne recherche que les mines enterrées de type "0". Toutes les mines du champ contaminé sont traitées et les "trous" remplacés par des "1". Quand la totalité du champ a été sécurisé le signal par un Drapeau positionné à "1" et placé à droite de l'étendue de BITS traitée. (L'étendue du champ est de longueur quelconque.)



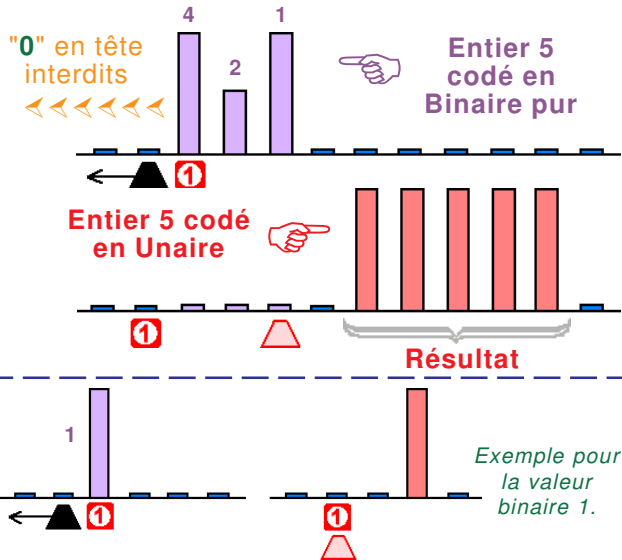
L'étendue traitée peut indifféremment commencer ou se terminer par une mine. La tête de L/E est dégagée à gauche.

Programme utilisateur n°13.

Réciproque du programme n°12 cet algorithme effectue le **Transcodage vers un Entier indiqué en Unaire d'une valeur exprimée en BINAIRE pur**. Les poids forts sont à gauche et il faut prévoir au moins N+1 espaces de type "B" à droite de la donnée.

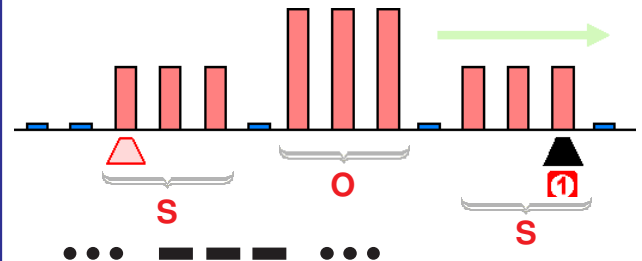
ATTENTION : Pas de "0" en tête, c'est à dire à gauche du poids fort.

(Pour la valeur 5 effectuée déjà 62 instructions.)



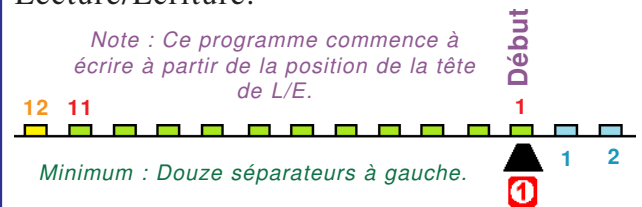
Programme utilisateur n°11.

Logiciel très simple dont l'algorithme positionne sur le barillet le **message SOS codé en Morse**. Les points sont représentés par des "0" et les traits par des "1". L'organisation des instructions est linéaire et l'exploration des TRANSITIONS se fait dans l'ordre de la n°1 à la n°11.



Pour obtenir un résultat "dégagé" il importe de prévoir deux pions de type "B" à droite de la tête de Lecture/Ecriture et minimum douze séparateurs "B" à gauche de la tête de Lecture/Ecriture.

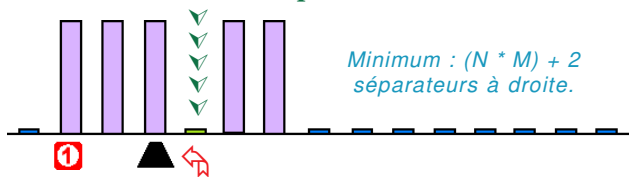
Note : Ce programme commence à écrire à partir de la position de la tête de L/E.



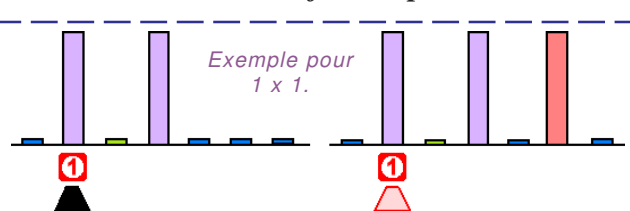
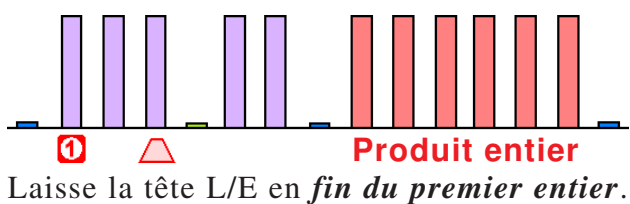
Programme utilisateur n°16.

Programme de calcul qui effectue la **multiplication de deux valeurs entières exprimées sous formes Unaires**.

ATTENTION : Pour la multiplication de 3 fois 2 la machine effectue déjà 118 cycles. **Un seul "B" doit séparer les deux entiers.**

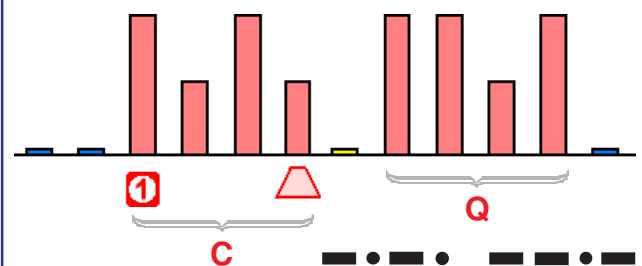


ATTENTION : La tête de lecture/Ecriture doit être à **Droite** de la première **Donnée**.



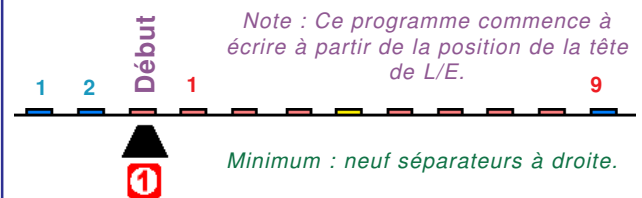
Programme utilisateur n°15.

Logiciel très simple dont l'algorithme positionne sur le barillet le **message d'appel CQ codé en Morse**. Les points sont représentés par des "0" et les traits par des "1". L'organisation des instructions est linéaire et l'exploration des TRANSITIONS se fait dans l'ordre de la n°1 à la n°11.



Pour obtenir un résultat "dégagé" il importe de prévoir deux pions de type "B" à gauche de la tête de Lecture/Ecriture et minimum neuf séparateurs "B" à droite de la tête de Lecture/Ecriture.

Note : Ce programme commence à écrire à partir de la position de la tête de L/E.



Programme utilisateur n°18.

Grand classique en génie informatique, cet algorithme *réalise la Conjecture de SYRACUSE* découverte par Collatz en 1930.

Prendre un nombre entier **N** :

- Si **N** est PAIR, *le diviser par 2.*
- Si **N** est IMPAIR, prendre $(3 \times N) + 1$.
- Recommencer avec les entiers successifs obtenus. La conjecture de Collatz prédit qu'à la fin, on obtient toujours 1.

Depuis 1930, bon nombre de mathématiciens cherchent à expliquer pourquoi cette conjecture est vraie, mais aujourd'hui personne n'y est encore parvenu. (Tout au moins en 2022.)

EXEMPLES :

3 \Rightarrow 10 \Rightarrow 5 \Rightarrow 16 \Rightarrow 8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1

4 \Rightarrow 2 \Rightarrow 1

5 \Rightarrow 16 \Rightarrow 8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1

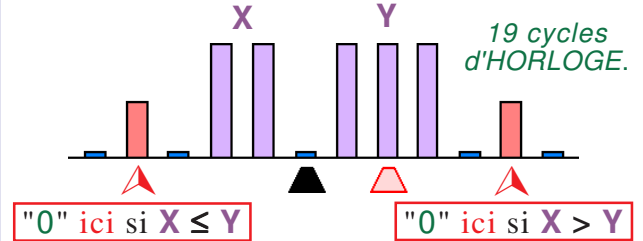
7 \Rightarrow 22 \Rightarrow 11 \Rightarrow 34 \Rightarrow 17 \Rightarrow 52 \Rightarrow 26 \Rightarrow 13 \Rightarrow 40 \Rightarrow 20 \Rightarrow 10 \Rightarrow 5 \Rightarrow 16 \Rightarrow 8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1

9 \Rightarrow 28 \Rightarrow 14 \Rightarrow 7 \Rightarrow 22 \Rightarrow 11 \Rightarrow 34 \Rightarrow 17 \Rightarrow 52 \Rightarrow 26 \Rightarrow 13 \Rightarrow 40 \Rightarrow 20 \Rightarrow 10 \Rightarrow 5 \Rightarrow 16 \Rightarrow 8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1 ... / ...

Programme utilisateur n°20.

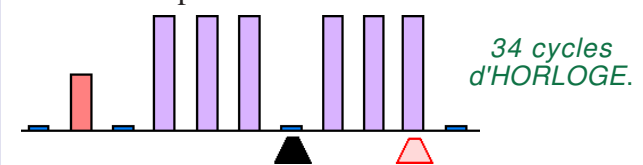
Réalise la comparaison entre deux Entiers codés sous forme Unaire.

Laisse la tête de Lecture/Ecriture dans une position variable à la fin du traitement. Un seul "B" sépare les données **X** et **Y** et la tête de Lecture/Ecriture doit initialement se trouver sous ce séparateur de **X** et de **Y**.



Comme précisé ci-dessus le résultat sera indiqué sous forme d'un drapeau "0" situé à gauche ou à droite des données **X** et **Y**.

Autre exemple avec **X = Y** :



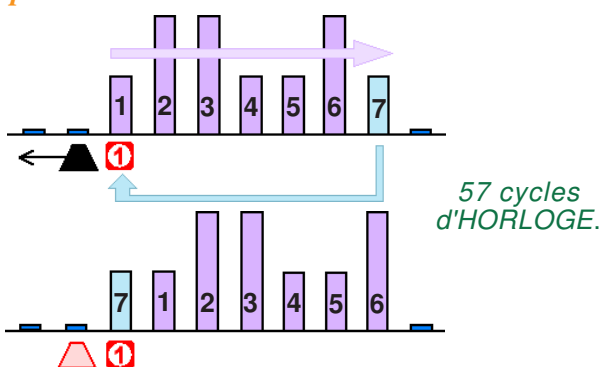
Si **X** = 3 et **Y** = 2 : 26 cycles d'HORLOGE.

Si **X** = 2 et **Y** = 4 : 19 cycles d'HORLOGE.

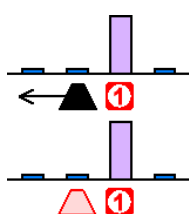
Si **X** = 4 et **Y** = 2 : 26 cycles d'HORLOGE.

Programme utilisateur n°21.

Réalise la permutation circulaire à Droite d'une donnée exprimée en Binaire pur. Laisse la tête de L/E à gauche du résultat final. La donnée binaire initiale peut présenter une taille quelconque, mais le nombre de cycles d'Horloge augmente rapidement avec le nombre de BITS.



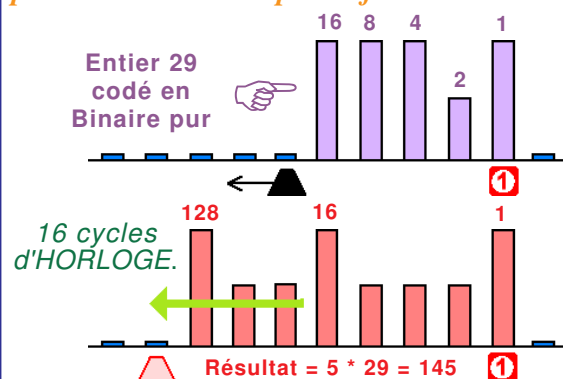
Cas particulier : Donnée a un seul BIT :



La configuration finale est strictement identique à l'organisation initiale après 15 cycles d'Horloge.

Programme utilisateur n°24.

Calcule cinq fois l'entier **N** exprimé en Binaire pur. Laisse la tête de L/E à gauche du résultat final. La donnée binaire initiale peut présenter une taille quelconque, mais il faut prévoir un espace de "B" suffisant situé à gauche de la donnée pour pouvoir écrire les poids forts du résultat.



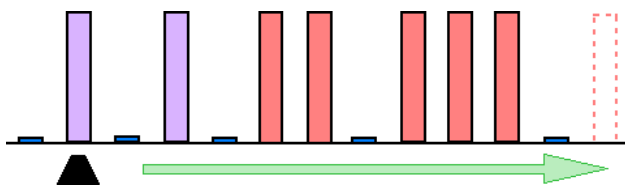
Programme utilisateur n°19.

A lgorithme qui *inscrit sous forme Unaires sur la machine la suite de FIBONACCI*. Logiquement l'ensemble du barillet doit être initialisé avec des "B". Les deux premiers termes sont chacun égaux à un et la tête de Lecture/Ecriture doit se trouver sous celui de gauche.

$$U_{n+2} = U_n + U_{n+1}$$

1 - 1 ⇒ 2 ⇒ 3 ⇒ 5 ⇒ 8 ⇒ 13 ⇒ 21 ⇒ 34 ...

Sur la machine on ne dispose au maximum d'une étendue de 56 positions. Il y aura perte par "écrasement" des données et divergence du programme à partir de l'élément 21 inclus.



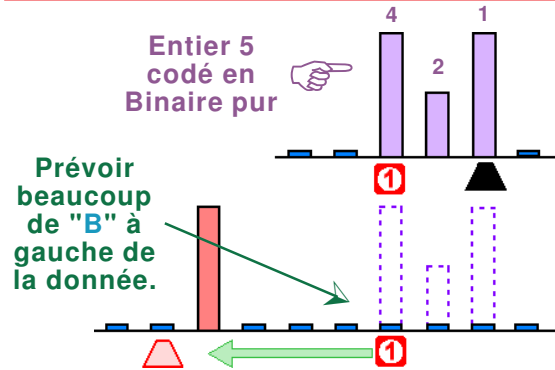
ATTENTION : Le nombre de cycles augmente très rapidement pour tracer les termes suivants. Par exemple quand le terme 8 est inscrit sur le plateau, la machine a déjà effectué environ 401 cycles d'Horloge.

☠ Ce programme est sans FIN. ☢

Programme utilisateur n°18.

L entier initial **N** est codé en BINAIRE pur sur la machine, les poids faibles étant à droite de façon classique. Ne pas indiquer des "0" en tête de la donnée.

ATTENTION : La tête de lecture/Ecriture doit être à **Droite** de la première **Donnée**.



Le programme va écrire les nombres successifs de la suite en binaire, en décalant systématiquement le codage vers la gauche. Il faut impérativement **prévoir BEAUCOUP de place libre à gauche** de la donnée initiale.

Si $N > 7$ exige beaucoup de cycles.

Si $N = 5$ engendre 18 cycles sur la machine.

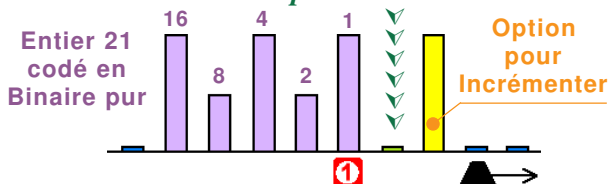
Si $N = 7$ engendre 79 cycles sur la machine.

Programme utilisateur n°23.

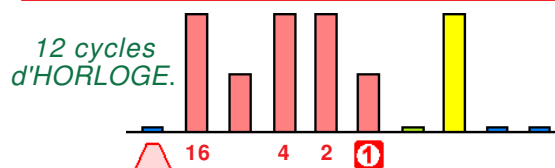
I ncrémenter / Décrémenter d'une unité une donnée codée en Binaire en fonction d'un BIT de sélection. Ce BIT d'option est situé à droite de la donnée Binaire séparé par un seul BIT de type "B".

- Sélecteur = "0" : **Décrém**ente de 1.
- Sélecteur = "1" : **Incrém**ente de 1.

Un seul "B" doit séparer les deux entiers.



ATTENTION : La tête de lecture/Ecriture doit être à **Droite** du BIT d'Option.



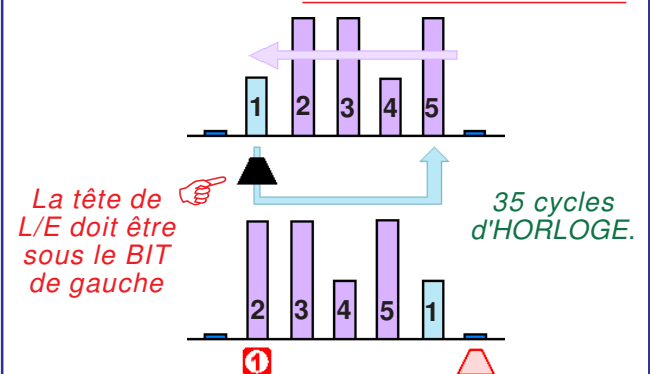
AUTRES EXEMPLES :

- 21 - 1 : 11 cycles.
- 22 + 1 : 12 cycles.

Programme utilisateur n°22.

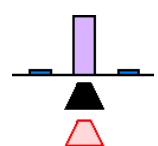
A nalogue au programme n°21, il effectue la permutation circulaire à Gauche d'une donnée exprimée en Binaire pur. Laisse la tête de L/E à droite du résultat final. La donnée binaire initiale peut présenter une taille quelconque, mais **le nombre de cycles d'Horloge augmente rapidement avec son nombre de BITS**.

>>> La tête L/E est sous le BIT de Gauche.



Cas particulier : Donnée a un seul BIT :

La configuration finale est strictement identique à l'organisation initiale après 10 cycles d'Horloge.

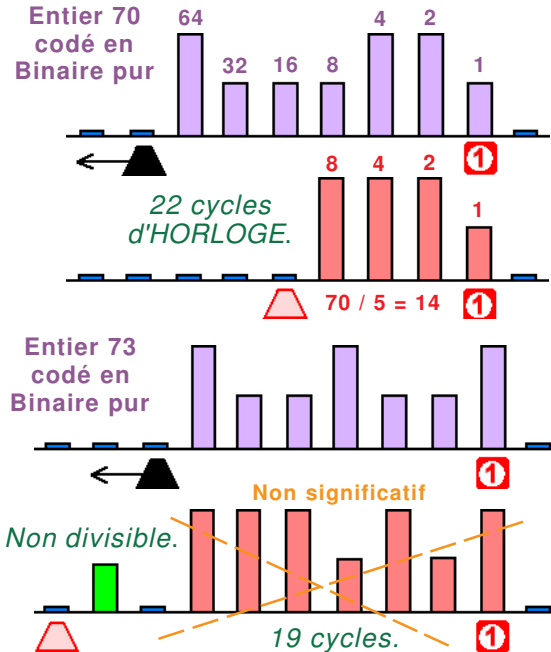


Programmes disponibles.

- PGM n°01 :** Écriture de "1" ou de "0".
- PGM n°02 :** • SHIFT à Droite de "1".
- Tête L/E placée en fin de 2^{ème} donnée.
- PGM n°03 :** • Inverser les BITS.
- Chercher [01] dans une chaîne.
- PGM n°04 :** Concaténer deux CHs de "1".
- PGM n°05 :** Parité du nombre de "1".
- PGM n°06 :** Multiplication Unaire par 2.
- PGM n°07 :** • GRAI vers BINAIRE.
- Conversion BINAIRE vers GRAI.
- PGM n°08 :** • Démineur "0" ou "1".
- PGM n°09 :** • ADDITION Unaire.
- SOUSTRACTION Unaire.
- PGM n°10 :** Élever 2 à la PUISSANCE *N*.
- PGM n°11 :** Écrire **SOS** en code Morse.
- PGM n°12 :** Coder un Entier vers Binaire.
- PGM n°13 :** Coder un Binaire vers Entier.
- PGM n°14 :** Diviser un Entier par trois.
- PGM n°15 :** Écrire l'appel **CQ** en Morse.
- PGM n°16 :** Multiplier deux entiers.
- PGM n°17 :** Déminer totalement les "0".
- PGM n°18 :** Conjecture de SYRACUSE.
- PGM n°19 :** Suite de FIBONACCI.
- PGM n°20 :** Comparaison de deux Entiers.

Programme utilisateur n°25.

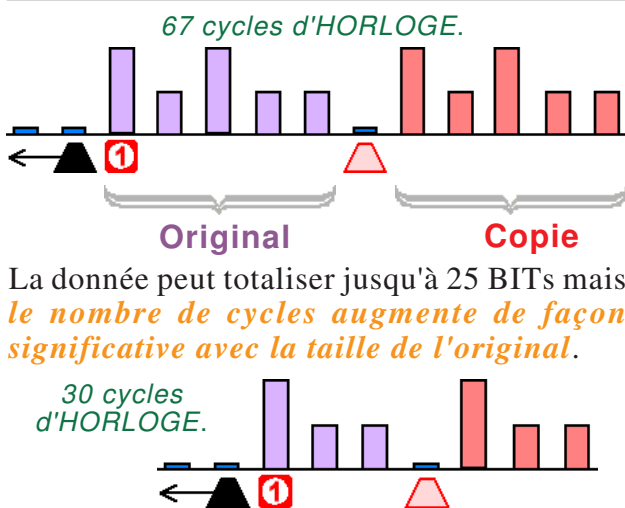
Calcule la division par cinq de l'Entier *N* exprimé en Binaire pur. Laisse la tête de L/E à gauche du résultat final. La donnée binaire initiale peut présenter une taille quelconque. *Si le résultat n'est pas un entier, positionne un indicateur à "0" à gauche de la donnée initiale.*



Programme utilisateur n°29.

Très orienté programmation binaire de microcontrôleurs, *ce logiciel duplique une chaîne de caractères* de longueur quelconque *écrite Binaire pur*, opération indispensable quand on désire mémoriser une valeur sur laquelle on va effectuer des modifications propres au traitement global.

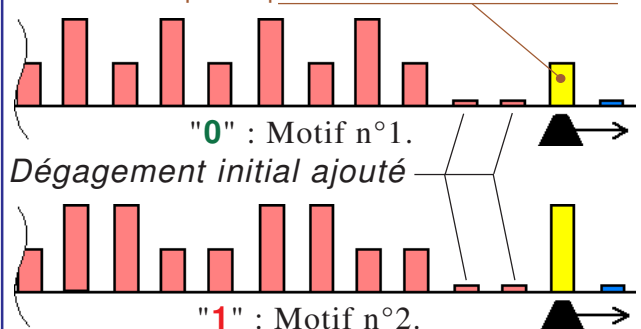
Il faut prévoir (*N*+5) espaces de type "B" à droite de la donnée, *N* étant le nombre de BITS caractérisant la chaîne à recopier.



Programme utilisateur n°28.

Particulièrement bien adapté pour effectuer sur la machine des test d'endurance, ce programme qui propose deux options se contente de *construire une frise sans FIN sur le barillet*. Durant son déroulement le plateau effectue sans interruption des rotations à Droite pour "dégager" le résultat. *L'aiguillage se produit sur le premier BIT différent de "B" rencontré*. Efface le BIT de sélection et ajoute deux autres "B" pour créer un dégagement initial en tête de la frise.

Sélection par le premier NON "B" rencontré.

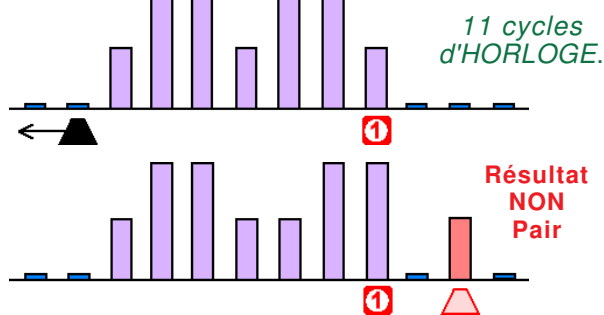
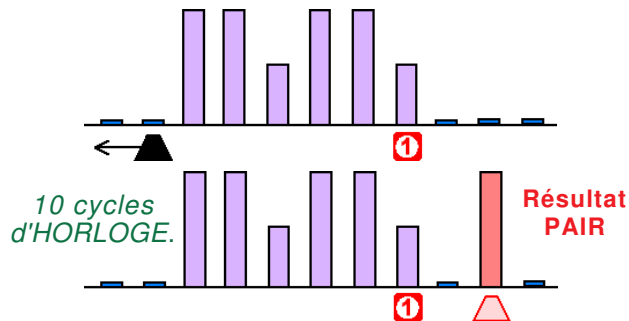


Ce programme fonctionne en "aveugle" et construit la séquence quel que soit l'état actuel des pions sur le barillet.

Programme utilisateur n°26.

Vérifie si le nombre de "0" est PAIR ET le nombre de "1" est PAIR sur une donnée **N** binaire. Laisse la tête de L/E à gauche du BIT traduisant le résultat :

"0" : Le Nbre de "0" ou de "1" n'est pas Pair.
 "1" : Le Nbre de "0" ET de "1" est PAIR.



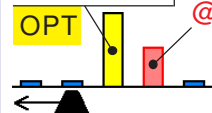
Programmes disponibles.

- PGM n°21 : Permutation circulaire à ➡.
- PGM n°22 : Permutation circulaire à ⬅.
- PGM n°23 : Incrémenter/Décrémenter d'une unité une donnée codée en Binaire.
- PGM n°24 : Calcule $5 * N$ en binaire.
- PGM n°25 : Calcule $N / 5$ en binaire.
- PGM n°26 : Vérifie si le nombre de "0" ET le nombre de "1" est PAIR dans un Binaire.
- PGM n°27 : Teste pour un PALINDROME.
- PGM n°28 : Construction d'une FRISE.
- PGM n°29 : Duplique une chaîne Binaire.
- PGM n°30 : Simule un registre FIFO.

PGM TEST : Algorithme élémentaire qui permet de générer une ER 1 ou une ER 2.

"0" : ER 1

"1" : ER 2



@	Ecriture	Transitions	OPT
1	B et 1		"0"
0	B et 0		
B	0 et 1		"1"
1		1 et 11	
0		3 et 9	
B		3, 6 et 7	

Remarque : PGM n°30 inclus : 133 trous.

NOTE : Pour imager la combinatoire explosive des algorithmes possibles, superposer les programmes 9, 19 et 27.

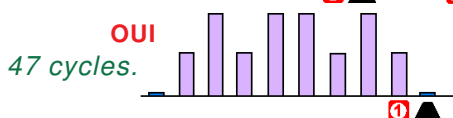
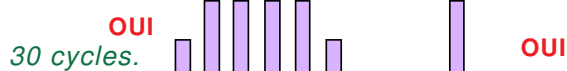
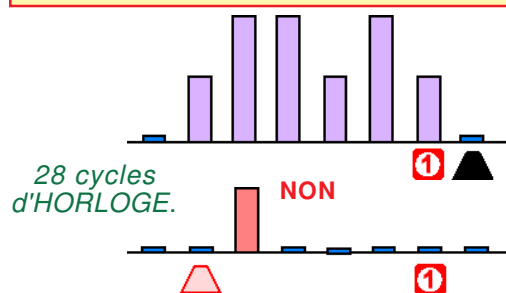
Programme utilisateur n°27.

Algorithme qui détecte si une donnée **N** écrite en Binaire pur constitue ou non un PALINDROME. Laisse la tête de L/E à gauche du BIT traduisant le résultat :

"0" : La donnée **N** n'est pas un palindrome.
 "1" : La donnée **N** constitue un palindrome.

Efface la donnée initiale **N**.

ATTENTION : La tête de lecture/Ecriture doit être immédiatement à Droite de **N**.

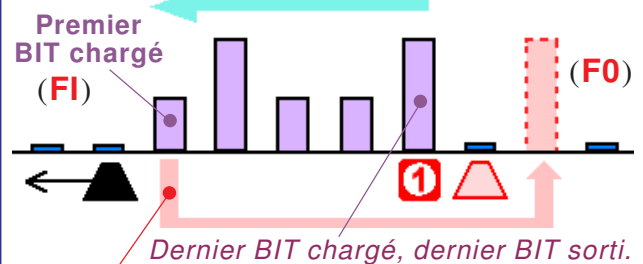


Programme utilisateur n°30.

Typique des opérations effectuées en binaire sur les microcontrôleurs, ce logiciel simule le fonctionnement d'un **Registre FIFO**. Ces opérations matérialisent des files d'attente en mémoire. Dans cet algorithme le registre Binaire est supposé avoir été chargé de la Droite vers la Gauche, le BIT d'écriture et de lecture étant à Droite. Il faut prévoir 3 à 4 "B" à droite du Registre.

FIFO : First IN, Firs OUT.

Sens supposé de chargement du Registre



Le premier BIT chargé d'initiales FI est le premier BIT sorti d'initiales FO.

Pour simuler le fonctionnement d'un registre FIFO, quand le dernier BIT a été présenté en sortie, le BIT de transfert est effacé à "B" pour simuler sa lecture suivi de sa purge.

Programmes disponibles.

PGM n°31 : ☐ Deux OPT : Castor Affairé.
PGM n°32 : ☐ Deux OPT : Castor Affairé.
PGM n°33 : ☐ Créer une clôture.
PGM n°34 : ☐ Construire un mur.
PGM n°35 : Calcule $5 * N + 1$ en BINAIRE.
PGM n°36 : Calcule $5 * N + 2$ en BINAIRE.
PGM n°37 : Calcule $5 * N + 3$ en BINAIRE.
PGM n°38 : Calcule $5 * N + 4$ en BINAIRE.
PGM n°39 : Calcule $N / 3$ en BINAIRE.
PGM n°40 : Créer la suite des ENTIERS.
PGM n°41 : ☐ Compteur / Décompteur.
PGM n°42 : Déplace une donnée BINAIRE.
PGM n°43 : Créer la suite des ENTIERS.
PGM n°44 : Divise un UNAIRE en deux.
PGM n°45 : ☐ Réalise une **Dichotomie**.
PGM n°46 : Page "artistique" **Sinusoïdes**.
PGM n°47 : Addition de deux BINAIRE.
PGM n°48 : Soustraction de deux " " " " ".
PGM n°49 : Recenser une population.
PGM n°50 : Compteur BINAIRE GRAY.
PGM n°51 : Décompteur BINAIRE GRAY.
PGM n°52 : Suite des puissances de deux.
PGM n°53 : PGCD de deux UNAIRE.
 Grille de base de 561 trous et tous les PGMs inclus représentent : **2699 trous fois deux**.

Programme utilisateur n°31.

Exemples de Castors affairés avec deux options choisies pour leur faible nombre de cycles. La zone concernée doit contenir au moins cinq "B" à gauche du pion de sélection et au moins quatre "B" à droite. Quand le programme est activé il faut que la tête de Lecture/Écriture soit au dessus du pion de sélection. Si cette position est un "B" il sera considéré comme un "0".

OPTION	États	Symboles	Cycles	Score
"0"	2	2	6 + 2	4
"1"	3	2	21 + 2	5

(Le +2 dans Cycles est issu de l'OPTION.)

Prévoir cinq à six "B" à gauche du BIT d'OPTION et quatre "B" à droite.

Déterminer un Castor Affairé est un problème non résolvable algorithmiquement. Une fonction Castor Affairé à partir d'un certain Score croît plus rapidement que n'importe quelle fonction calculable. En pratique, on ne peut même pas espérer traiter un castor affairé pour un Score au-delà de 10. (Voir la fiche du PGM n°32 qui détaille ce point.)

... / ...

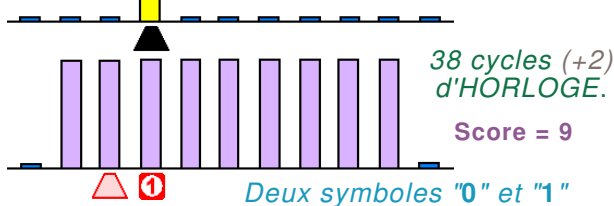
Programme utilisateur n°32.

PGM avec options pour deux Castors affairés.

Prévoir 3 "B" à gauche et 7 à droite.

Option pour deux États

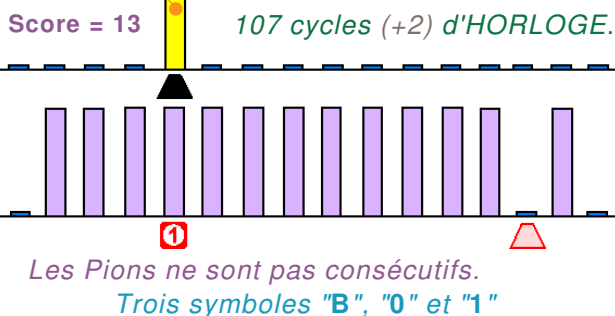
Exemples pour l'Option "0".



Prévoir 4 "B" à gauche et 11 à droite.

Option pour quatre États

Exemples pour l'Option "1".

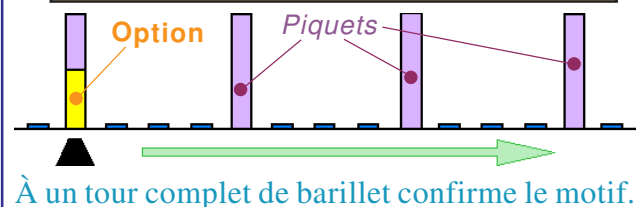


Programme utilisateur n°33.

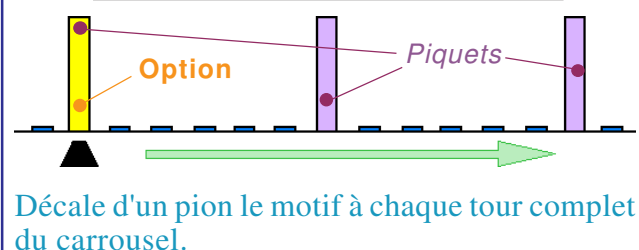
Créer un enclos quel que soit l'état des pions sur le carrousel au départ. Deux OPTIONS sont possibles en fonction de l'état du pion situé sous la tête de Lecture / Écriture. **Simule l'implantation d'une clôture par symbolisation de piquets espacés à égale distance les uns des autres.**

ATTENTION : Programmes sans Fin.

"B" ou "0" : 3 Espaces entre les piquets.



"1" : 5 Espaces entre les piquets.

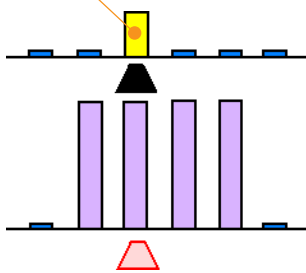


Programme utilisateur n°31.

Prévoir cinq à six "B" à gauche du BIT d'OPTION et quatre "B" à droite.

Option pour deux États

Exemples pour l'Option "0".



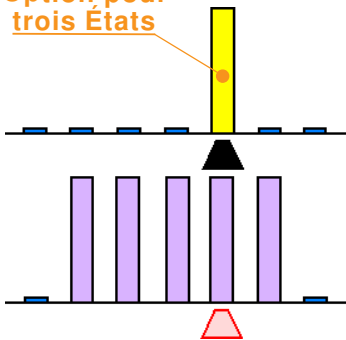
Deux symboles

6 cycles (+2)
d'HORLOGE.

Score = 4

Option pour trois États

Exemples pour l'Option "1".



Deux symboles

21 cycles (+2)
d'HORLOGE.

Score = 5

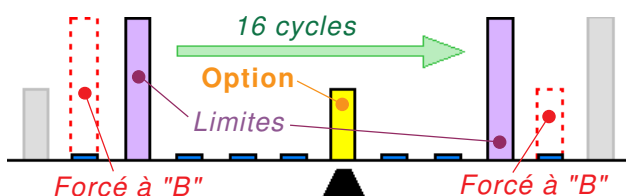
Les deux symboles sont "B" et "1"

Programme utilisateur n°34.

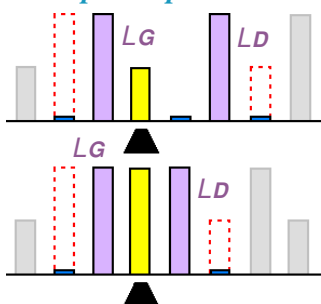
Construit un mur entre deux limites. Deux OPTIONS possibles en fonction de l'état du pion situé sous la tête de L / E :

- "B" ou "0" : Réalise de Gauche à Droite.
- "1" : Réalise de Droite à Gauche.

Force à "B" les pions de part et d'autre des limites, et termine sur le pion "effacé". À l'extérieur des Limites les états du plateau peuvent être strictement quelconques.



Bien que ce ne soit pas vraiment logique le pion d'Option peut être contre les limites :



Configuration illogique mais qui fonctionne.

Programme utilisateur n°32.

Ce concept introduit en 1962 de **castor affairé**, dont le nom (1) est proposé par le mathématicien hongrois Tibor Radó est l'un des 1^{er} **exemples de fonction non calculable**. Un castor affairé est, en théorie de la calculabilité, une machine de Turing qui maximise son activité opérationnelle comme le **nombre de pas effectués ou le nombre de symboles écrits avant son arrêt** : Son **Score**.

Ces classes de Machines de Turing doivent s'arrêter **après être lancées sur un ruban vierge** et respecter les particularités :

- La machine utilise un ruban unique,
- La longueur du ruban est illimitée,
- L'alphabet est {0, 1} avec le "0" servant servant de symbole vierge.
- La Transition dépend de deux entrées :
 - * L'ÉTAT actuellement actif,
 - * Le Symbole lu en début de cycle.

États	2	3	4	5	6
Cycles	6	21	107	47176870	7,4 x 10 ³⁶⁵³⁴

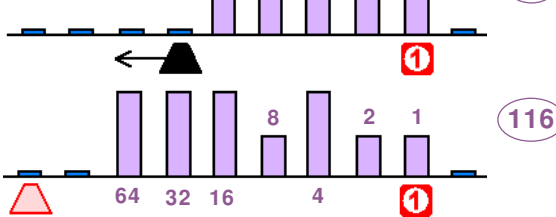
(1) "Busy beaver" : Désigne familièrement une personne besogneuse et travailleuse.

Programme utilisateur n°35.

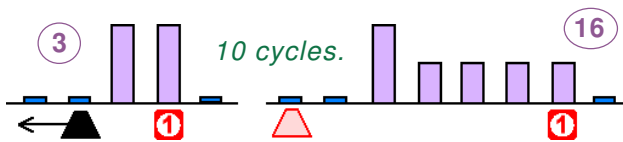
C *alcule $5 * N + 1$ en BINAIRE pur.* Peut calculer des valeurs vraiment très grandes. Maximum 2 élevé à la puissance 52 soit la bagatelle de 9.007.199.254.740.991 !
ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec la valeur initiale proposée pour **N**.

Prévoir un nombre suffisant de "B" à Gauche de la donnée initiale **N**.

15 cycles d'HORLOGE. (23)



Par exemple si **N** = 13.421.771 le résultat sera égal à la valeur binaire de 67.108.856 soit à peine 34 BITS modifiés sur le carrousel.

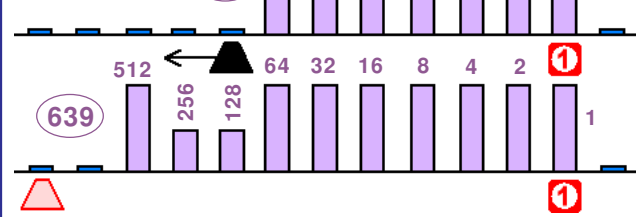


Programme utilisateur n°38.

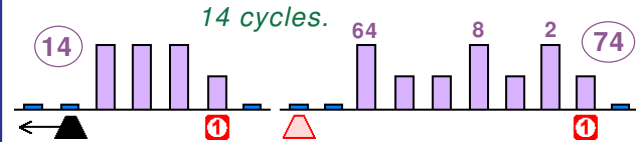
C *alcule $5 * N + 4$ en BINAIRE pur.* Peut calculer des valeurs vraiment très grandes. Maximum 2 élevé à la puissance 52 soit la bagatelle de 9.007.199.254.740.991 !
ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec la valeur initiale proposée pour **N**.

Prévoir un nombre suffisant de "B" à Gauche de la donnée initiale **N**.

20 cycles d'HORLOGE. (127)



Par exemple si **N** = 13.421.771 le résultat sera égal à la valeur binaire de 67.108.856 soit à peine 34 BITS modifiés sur le carrousel.

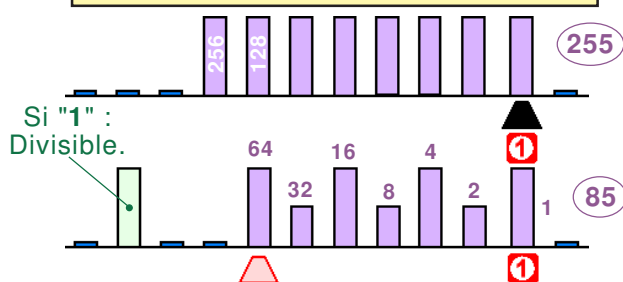


Programme utilisateur n°39.

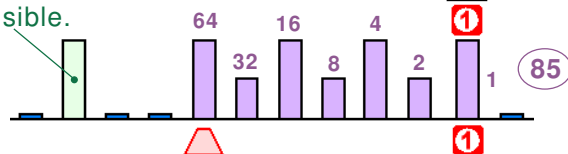
E *ffectue la division d'un nombre N par 3 exprimé en BINAIRE pur. Teste la divisibilité et supprime les zéros en tête.*

Note : Il peut y avoir un ou plusieurs séparateurs entre la fin de la donnée et l'indicateur de divisibilité.

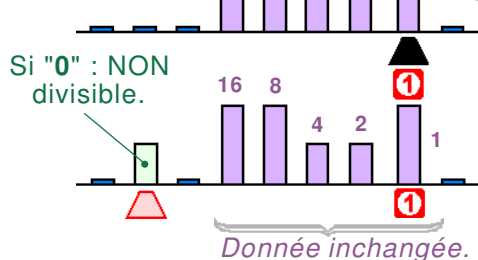
Prévoir 3 "B" à gauche et 1 à droite.



Si "1" : Divisible.



Si "0" : NON divisible.



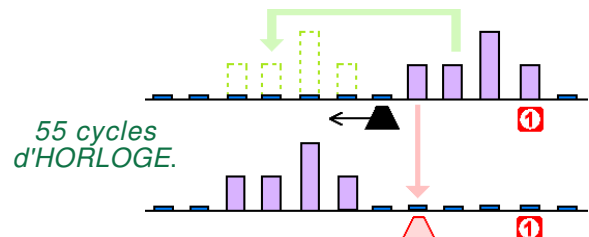
Donnée inchangée.

Programme utilisateur n°42.

D *éplace une donnée BINAIRE à gauche dans la mémoire vive de la machine.* Le décalage se fait à deux BITS à gauche du poids fort de la donnée initiale qui est effacée de la mémoire. Sur un ordinateur actuel une telle donnée serait déplacée d'une cellule mémoire au format 8, 16, 32 ou 64 BITS. Sur la machine de Turing dont la mémoire est non formatée il y a optimisation de l'espace mémoire, le décalage de la transposition est minimalisé.

Prévoir autant de "B" à gauche de la donnée qu'elle compte de BITS pour sa définition initiale **plus trois espaces**.

Laisse la tête de L / E sur le BIT de poids fort effacé de la donnée initiale.

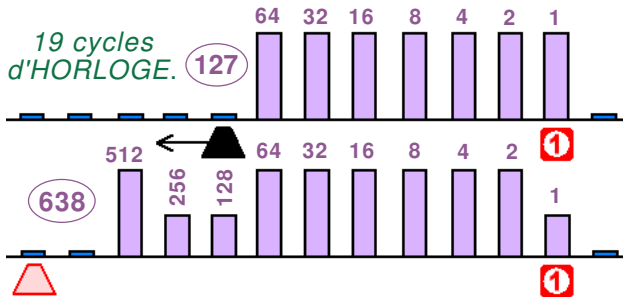


ATTENTION : Le nombre de cycles pour l'HORLOGE augmente très rapidement avec le nombre de BITS de la donnée BINAIRE.

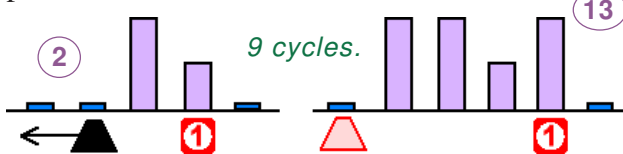
Programme utilisateur n°37.

C *alcule $5 * N + 3$ en BINAIRE pur.* Peut calculer des valeurs vraiment très grandes. Maximum 2 élevé à la puissance 52 soit la bagatelle de 9.007.199.254.740.991 !
ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec la valeur initiale proposée pour **N**.

Prévoir un nombre suffisant de "B" à Gauche de la donnée initiale **N**.



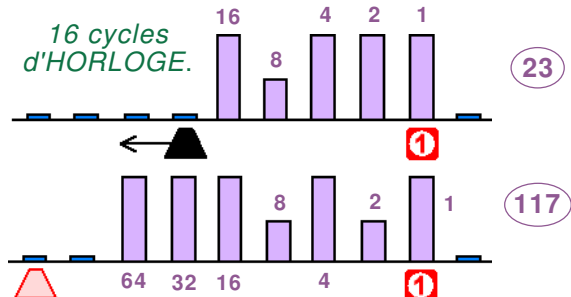
Par exemple si **N** = 13.421.771 le résultat sera égal à la valeur binaire de 67.108.856 soit à peine 34 BITS modifiés sur le carrousel.



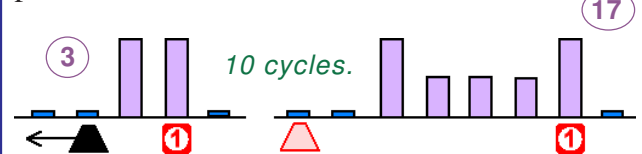
Programme utilisateur n°36.

C *alcule $5 * N + 2$ en BINAIRE pur.* Peut calculer des valeurs vraiment très grandes. Maximum 2 élevé à la puissance 52 soit la bagatelle de 9.007.199.254.740.991 !
ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec la valeur initiale proposée pour **N**.

Prévoir un nombre suffisant de "B" à Gauche de la donnée initiale **N**.



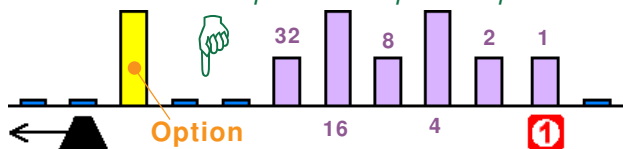
Par exemple si **N** = 13.421.771 le résultat sera égal à la valeur binaire de 67.108.856 soit à peine 34 BITS modifiés sur le carrousel.



Programme utilisateur n°41.

R *éalise un Compteur / Décompteur prépositionnable.* Peut être initialisé à **N** > 0 même en OPTION de Comptage. *Il faut au moins un BIT de donnée à "0" ou à "1".*

Nombre de séparateurs quelconque



Option = "0" : Décomptage.

Décompte la valeur prépositionnée jusqu'à zéro puis efface les "0". La tête de L / E termine le PGM sous l'origine initiale **1**.

Exemple : **N** = 6 : Décomptage en 42 cycles.

Option = "1" : Comptage.

Augmente la *valeur BINAIRE* initialisée *de façon infinie*. (Pas de Fin de programme.)

Il ne faut que des "B" sur le plateau au clic sur le B.P. **AUTOM.** sauf pour l'OPTION et pour la *donnée codée en BINAIRE pur*.

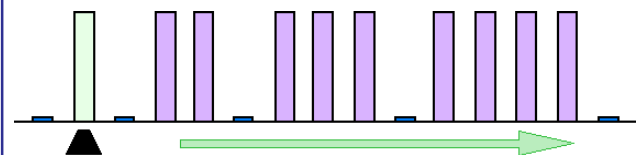
ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec la valeur de **N** en cours d'incrément.

Programme utilisateur n°40.

C *onstruit la suite des entiers naturels codés en UNAIRE.* La machine recopie à droite l'élément **n** et ajoute un pion à "1".

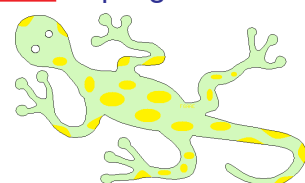
Attention, ce programme est sans fin et impose pour sa sortie une coupure alimentation hors de la fonction LECTURE sur l'HORLOGE.

Il ne faut que des "B" sur le plateau au lancement du programme sauf sous la tête de Lecture/Écriture où l'on doit positionner un "1".



Le temps d'exécution augmente régulièrement avec le nombre de pions à recopier de l'élément **n** sur le plateau à l'élément **n+1**.

Préférer le programme n°43.

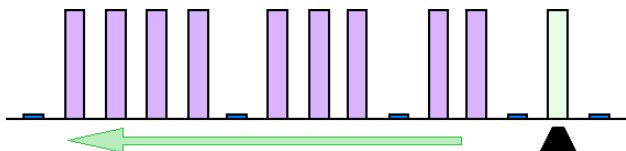


Programme utilisateur n°43.

Construit la suite des entiers naturels codés en **UNAIRE**. La machine recopie à droite l'élément n et ajoute un pion à "1".

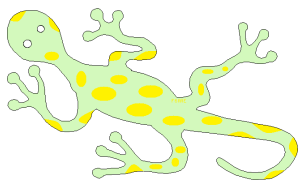
Attention, ce programme est sans fin et impose pour sa sortie une coupure alimentation hors de la fonction LECTURE sur l'HORLOGE.

Il ne faut que des "B" sur le plateau au lancement du programme sauf sous la tête de Lecture/Écriture où l'on doit positionner un "1".



Le temps d'exécution augmente régulièrement avec le nombre de pions à recopier de l'élément n sur le plateau à l'élément $n+1$.

Préférable au programme n°40.



Programme utilisateur n°45.

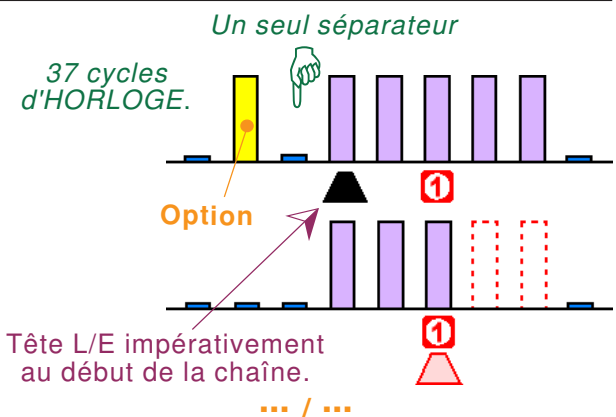
Réalise une **DICHOTOMIE** sur une chaîne **UNAIRE IMPAIRE**. En fonction du positionnement d'un BIT d'**Option** la zone effacée sera à gauche ou à droite de la donnée. Le centre de la chaîne **UNAIRE** sera positionné de préférence sur une origine de type ①.

Option = "0" effacement à Gauche.

Option = "1" effacement à Droite.

La chaîne **UNAIRE** doit comporter au moins trois BITS à "1" et la tête de L/E doit être initialisée à gauche de la donnée.

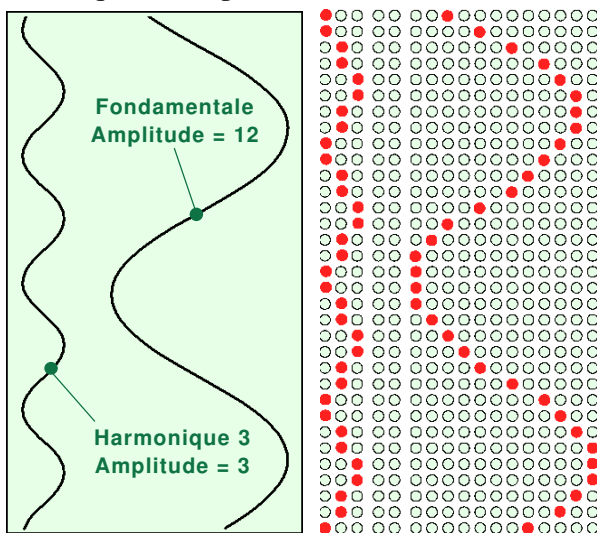
Il faut un seul séparateur entre le BIT d'**Option** et les éléments de la chaîne.



Programme utilisateur n°46.

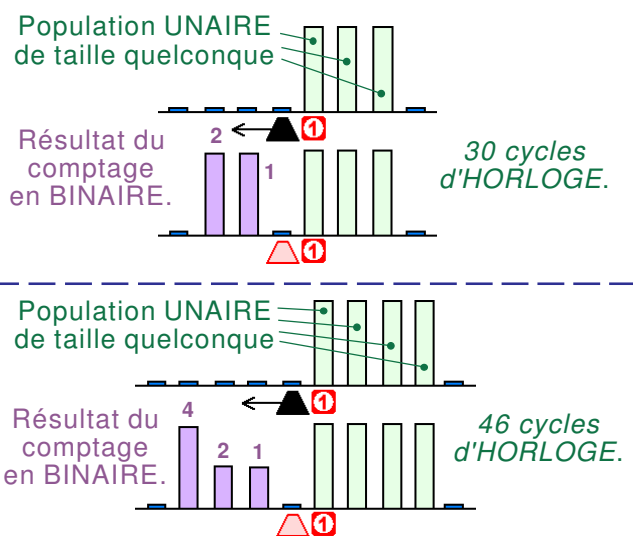
Contrairement à un **Castor Affairé**, cet algorithme avec 66 instructions ne génère que quatre cycles d'HORLOGE pour se terminer. En outre, il inclut dix écritures surabondantes et 29 lignes de transitions inutiles non explorées par le programme.

Cette fonction NON CALCULABLE concrétise une "fantaisie artistique" qui utilise les trous de la feuille perforée pour créer le dessin géométrique de deux ondes sinusoïdales harmoniques d'amplitudes différentes.



Programme utilisateur n°49.

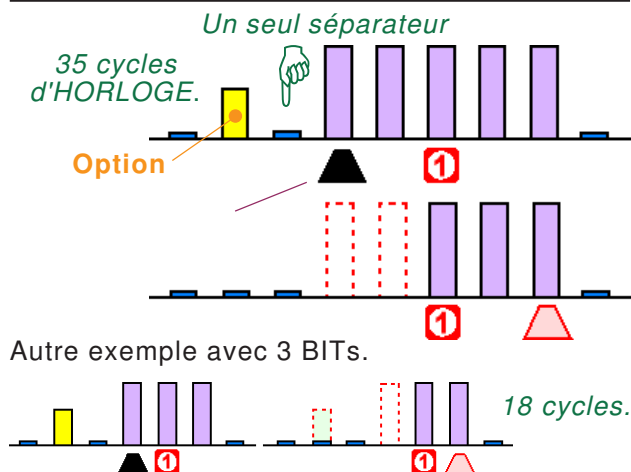
Répresentatif des premières tabulatrices électromécaniques utilisées aux USA pour les recensements ou le dépouillement des élections, ce programme simule un **comptage BINAIRE** pour recenser une population **UNAIRE**. Prévoir un nombre de "B" suffisants à gauche pour le compteur **BINAIRE**.



ATTENTION : Le nombre de cycles d'HORLOGE augmente très rapidement avec le nombre d'individus dans la population.

Programme utilisateur n°45.

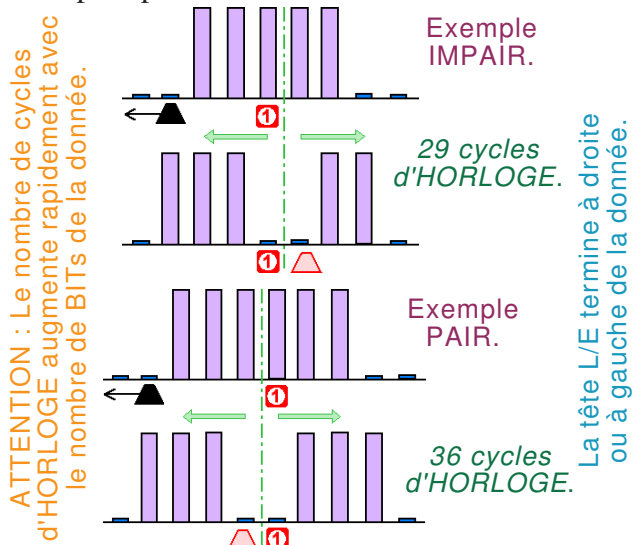
Efface le BIT d'**Option** en fin de traitement et laisse la tête de L/E sous l'origine ① si effacement à Droite, ou à Droite du résultat de la dichotomie si effacement à Gauche.



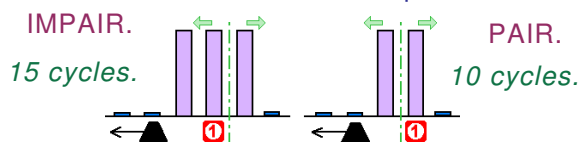
NOTE : En botanique, la dichotomie est la division d'un organe végétal en deux parties. En mathématique ou en informatique une dichotomie consiste à éliminer la moitié des éléments d'un ensemble quelconque "à sa droite" ou "à sa gauche".

Programme utilisateur n°44.

Sépare une chaîne de longueur quelconque en deux "moitiés" **UNAIRES**. Si le nombre de BITS de la donnée est **IMPAIR** la "moitié la plus grande" sera décalée à gauche et la plus petite à droite.



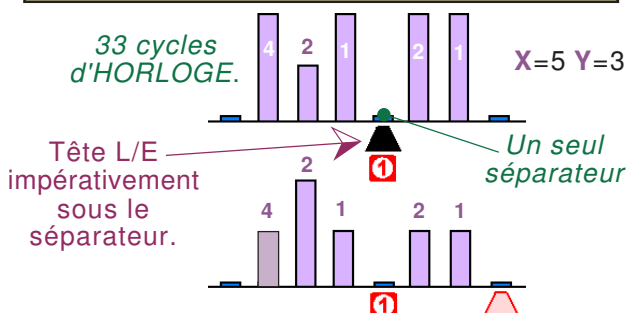
Deux autres exemples.



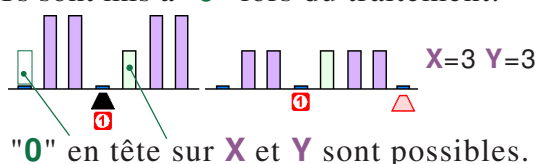
Programme utilisateur n°48.

Soustraction de deux **BINAIRES**. Calcule la valeur de **X** en faisant $X - Y$. En fin de programme **Y** est forcée à zéro, les données étant toutes codées en **BINAIRE** pur.

- La tête de L/E doit impérativement se trouver sous la séparation de **X** et **Y**.
- Un seul "**B**" pour séparer **X** et **Y**.
- X** doit être supérieur ou égal à **Y**.



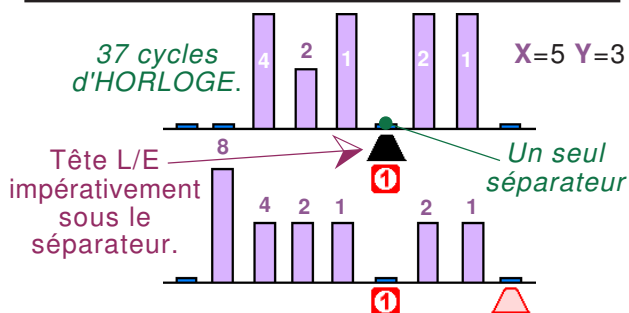
La valeurs de **Y** peut être nulle. La tête de L/E termine à droite de **Y** dont tous les BITS sont mis à "0" lors du traitement.



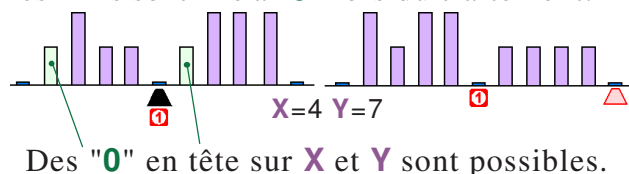
Programme utilisateur n°47.

Addition de deux **BINAIRES**. Calcule la valeur de **X** en faisant la somme $X + Y$. En fin de programme **Y** est forcée à zéro, les données étant toutes codées en **BINAIRE** pur.

- La tête de L/E doit impérativement se trouver sous la séparation de **X** et **Y**.
- Un seul "**B**" pour séparer **X** et **Y**.
- Prévoir assez de "**B**" à gauche de **X**.



L'une des deux valeurs initiale peut être nulle. La tête de L/E termine à gauche de **Y** dont tous les BITS sont mis à "0" lors du traitement.

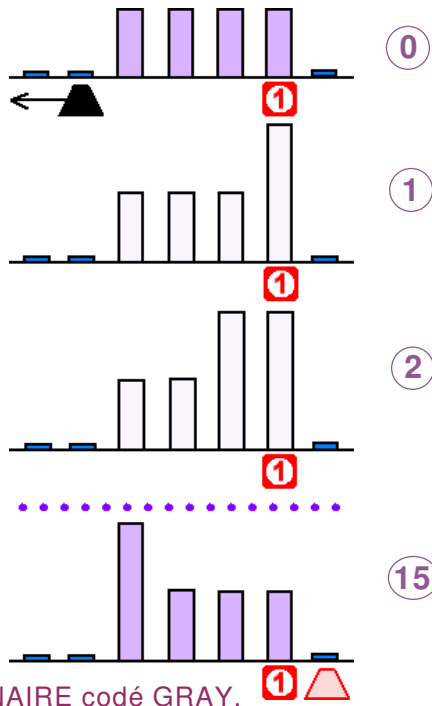


Programme utilisateur n° 50.

Réalise un **Compteur BINAIRE codé en GRAY**. Débute à la valeur zéro codée en GRAY comme en BINAIRE pur. Termine à la valeur quinze qui en code GRAY s'écrit "1000".

ATTENTION : 140 cycles d'HORLOGE.

Voir le dos de la Fiche n°52 pour le concept de déroulement de cet algorithme.



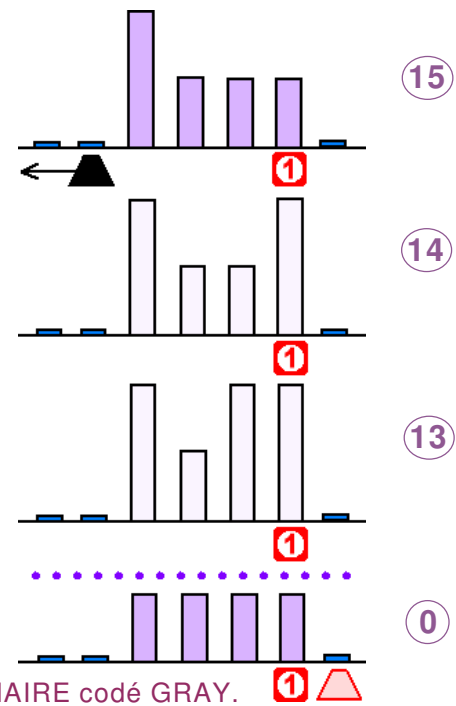
15 en BINAIRE codé GRAY.

Programme utilisateur n° 51.

Réalise un **Décompteur BINAIRE codé en GRAY**. Débute à la valeur 15 codée en GRAY qui s'écrit "1000". Termine à la valeur zéro qui s'écrit comme en BINAIRE pur.

ATTENTION : 140 cycles d'HORLOGE.

Voir le dos de la Fiche n°52 pour le concept de déroulement de cet algorithme.



0 en BINAIRE codé GRAY.

Programme utilisateur n° 50 & 51.

Elémentaires à décrire, ces programmes imposent un très grand nombre de cycles d'HORLOGE car il doivent effectuer régulièrement des déplacements de la tête de L/E à droite et à gauche de la donnée ce qui impose une foultitude de fonctions lecture.

➤ Algorithme pour le COMPTEUR.

La donnée de départ est initialisée à "0000".

- Si le nombre actuel de "1" est PAIR alors inverser le BIT de poids faible à Droite.
- Si le nombre actuel de "1" est IMPAIR alors inverser le BIT situé à Gauche du BIT "1" se trouvant le plus à Droite de la donnée.
- Si la valeur actuelle vaut "1000" : **FIN**.

➤ Algorithme du DÉCOMPTEUR.

La donnée de départ est initialisée à "1000".

- Si le nombre actuel de "1" est IMPAIR alors inverser le BIT de poids faible à Droite.
- Si le nombre actuel de "1" est PAIR alors inverser le BIT situé à Gauche du "1" se trouvant le plus à Droite de la donnée.
- Si la valeur actuelle vaut "0000" : **FIN**.

Programme utilisateur n° 53.

Pour déterminer le **PGCD** des deux nombres **X** et **Y** le programme utilise l'**algorithme d'Euclide** qui se résume à :

- Si $X < Y$ alors enlever X à Y .
- Si $X > Y$ alors enlever Y à X .
- Recommencer tant que l'un des deux n'est pas égal à zéro. Si c'est le cas $X = \text{PGCD}$.

Déroulement de l'algorithme.

- On force à "0" le "1" situé à Droite de **X**. On teste s'il reste encore un "1" à Gauche. Si ce n'est pas le cas on saute en (D). Si c'est le cas on saute en (B).
- On force à "0" le "1" situé à Gauche de **Y**. On teste s'il reste encore un "1" à Droite. Si ce n'est pas le cas $Y = 0$ on va en (C). Si c'est le cas on recommence en (A).
- On remet tous les BITS de "**Y**" à "1", puis On revient sur **X**, on efface **Y** BITS. On teste s'il reste encore un "1" à Gauche. Si ce n'est pas le cas on saute en (E). Si c'est le cas on revient en (A).
- Traitement comme en (E) mais les rôles de de **X** et de **Y** sont inversés et **G** devient **D**.
- La tête de L/E va à Droite de **X** et **FIN**.

Programme utilisateur n°51.

N	Binaire pur	Code GRAI
15	1111	1000
14	1110	1001
13	1101	1011
12	1100	1010
11	1011	1110
10	1010	1111
09	1001	1101
08	1000	1100
07	0111	0100
06	0110	0101
05	0101	0111
04	0100	0110
03	0011	0010
02	0010	0011
01	0001	0001
00	0000	0000

Programme utilisateur n°50.

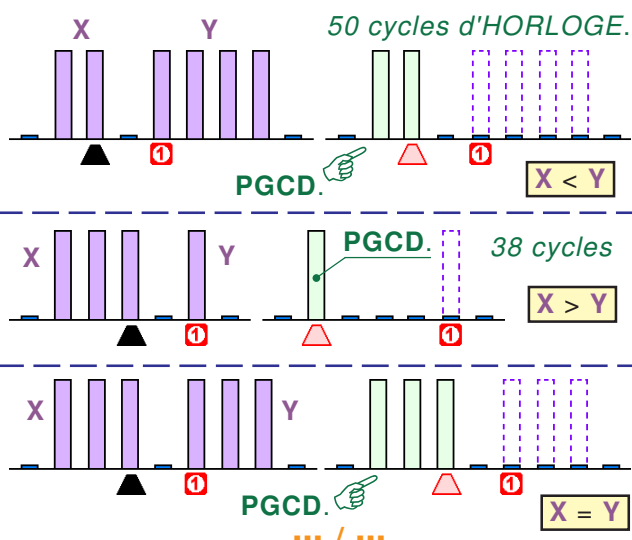
N	Binaire pur	Code GRAI
00	0000	0000
01	0001	0001
02	0010	0011
03	0011	0010
04	0100	0110
05	0101	0111
06	0110	0101
07	0111	0100
08	1000	1100
09	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Programme utilisateur n°53

Déterminer le PGCD de deux UNAIRES X et Y quelconques. La donnée X est remplacée par le résultat également codé en UNAIRE. La donnée Y est effacée par des "B". Programme achevé la tête de L/E se trouve à droite du résultat. *En début de programme la tête de L/E doit se trouver à droite de X .*

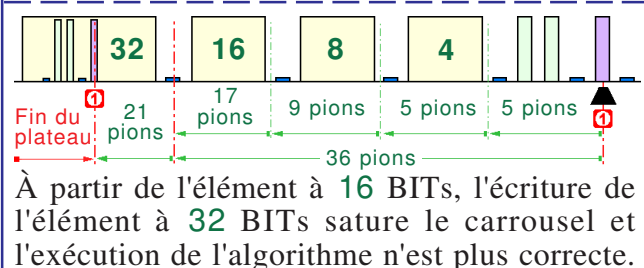
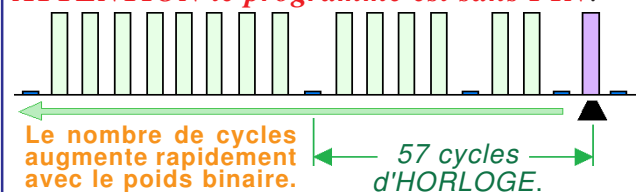
X peut être inférieur, égal ou supérieur à Y .

Le nombre de cycles d'HORLOGE augmente rapidement avec la taille de X ou (et) de Y .



Programme utilisateur n°52.

Construit la suite des puissances de deux en UNAIRES. L'ensemble du barillet ne doit contenir que des "B" sauf sous la tête de Lecture/Écriture où doit se trouver un "1". **ATTENTION le programme est sans FIN.**



Principe de l'algorithme :
On duplique n en $n+1$ à gauche en forçant les BITS de n à "0". Quand tous les BITS de n sont à "0" on recopie une deuxième fois des "1" à gauche de $n+1$ en restituant les BITS de n à "1". Lorsque tous les bits de n sont à "1" on revient sur $n+1$ qui se change en n et on recommence.