Coulonval Benjamin

Terminal - S

~

Suite de Skolem

Approche théorique du problème : créé un programme générant toutes suites de Skolem pour un ordre « n » équivalent à huit.

→ *Avant propos* : L'ensemble des fonctions utilisées en C++ dans le cadre de la réalisation du programme seront détaillées ligne par ligne dans le script en C++ joint à ce travail. Dans cette partie, il s'agira uniquement d'exposer la méthode élaboré permettant de trouver l'ensemble des suites de skolem pour un ordre équivalent à huit, soit 504 solutions. Enfin, dans le but de simplifier les explications, les exemples seront parfois expliqués pour une suites d'ordre 4, afin de minimisé la rédaction et optimiser la compréhension (La logique est équivalente selon l'ordre de la suite, donc pas de changement).

→ Mise en relation des paires de la suite avec leur positions :

Pour commencer, je propose que l'on observe cette exemple de suite de *skolem* :

Position →	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
Suite →	3	5	6	3	8	4	5	7	6	4	1	1	8	2	7	2

Nous pouvons dès lors établir un lien entre la position d'une paires A, et des valeurs de leurs positions tel que : **A=Y-X** avec **A** → N° de la paire, dans l'intervalle : [1;8]
Y → N° de la position la plus grande d'une paire A
X → N° de la position la plus petite d'une paire A

→ A partir de cette *exemple de suite* et de cette f*ormule* trouvé précédemment entre positions et paire A, nous pouvons dès lors tracer un tableau tel que :

A	=	Y	-	X
1		12		11
2		16		14
3		4		1
4		10		6
5		7		2
6		9		3
7		15		8
8		13		5

/!\ → Une suite de skolem implique le fait que chaque position doit être occupé qu'une seul fois, et donc que chaque X et Y soit différent de tel manière à conforter l'équation suivante : **{X1, X2, X3, ...,X15, X16} = {1, 2, 3, …, 15, 16}**

→ de ce fait, le programme devra avoir pour but principale de trouver tout X et tout Y, de manière à ce que la différence Y-X fasse toutes les paires A (de 1 à 8), et que chaque X et Y soit différent entre eux. (Ainsi, nous ne travaillerons qu'avec les n° de positions dans le programme, et non les paires A, que nous n'afficherons qu'à la fin par la différence de Y par X).

→ Principe fondamentale de la méthode de recherche des suites :

Remarque → Il n'existe aucune formule connue à ce jours permettant de calculer intégralité des suites de skolem pour un ordre donné n. De ce fait, le programme devra faire des recherches successives (de manière non-aléatoire et réduite) et testé si la suite généré est une suite de skolem. Le principale but de la partie théorique du probléme est donc de trouver une manière de réduire l'espace de recherche du programme de façon à ce que celui-ci puisse trouver de manière efficace, (pour ne pas que le programme mette

un temps infini à trouver une suite). C'est pour cela qu'utiliser les position va nous être utile :

→ Dans un premier temps, il faut chercher dans un tableau toutes les manières d'écrire les paires A , avec un couple de position (Y ; X), qui je rappel sont dans l'intervalle [1;16]

A	1	2	3	4	5	6	7	8
1	16 15	16 14	16 13	16 12	16 11	16 10	16 9	16 8
2	15 14	15 13	15 12	15 11	15 10	15 9	15 8	15 7
3	14 13	14 12	14 11	14 10	14 9	14 8	14 7	14 6
4	13 12	13 11	13 10	13 9	13 8	13 7	13 6	13 5
5	12 11	12 10	12 9	12 8	12 7	12 6	12 5	12 4
6	11 10	11 9	11 8	11 7	11 6	11 5	11 4	11 3
7	10 9	10 8	10 7	10 6	10 5	10 4	10 3	10 2
8	9 8	9 7	9 6	9 5	9 4	9 3	9 2	9 1
9	8 7	8 6	8 5	8 4	8 3	8 2	8 1	8
10	7 6	7 5	7 4	7 3	7 2	7 1	7	7
11	6 5	6 4	6 3	6 2	6 1	6	6	6
12	5 4	5 3	5 2	5 1	5	5	5	5
13	4 3	4 2	4 1	4	4	4	4	4
14	3 2	3 1	3	3	3	3	3	3
15	2 1	2	2	2	2	2	2	2
16	1	1	1	1	1	1	1	1

Y-X

y \ x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		1	2	3	4	5	6	7	8							
2			1	2	3	4	5	6	7	8						
3				1	2	3	4	5	6	7	8					
4					1	2	3	4	5	6	7	8				
5						1	2	3	4	5	6	7	8			
6							1	2	3	4	5	6	7	8		
7								1	2	3	4	5	6	7	8	
8									1	2	3	4	5	6	7	8
9										1	2	3	4	5	6	7
10											1	2	3	4	5	6
11												1	2	3	4	5
12													1	2	3	4
13														1	2	3
14															1	2
15																1
16																

X

→ *Méthode principale :* L'idée du programme est donc de faire une recherche successive de toutes les combinaisons possible **entre les différents** couples (Y;X), des différentes paires A (1 à 8) du premier tableau (le deuxième exprimant la même chose, mais de manière plus visible). C'est comme si nous allions programmer une horloge, sauf que, au lieu d'avoir les secondes, minutes et heures, nous nous aurons un couple pour former la paire A=8 puis un couple pour former A=7 ; … ; et pour A=1. Par exemple, le programme fera défiler tous les couples pour former la paireA=8, et une fois qu'ils aurons tous défilé (donc X sera égale à 1, la plus petite position), alors les variables du couple de position formant la paire A=7 ce décalerons de « -1 ».

→ *Remarque :* Pour passer du couple (16;8) formant A=8 au suivant, il suffit d'appliqué « -1 » aux composante du couple et on obtient alors le couple (15;7), formant toujoursla paire A=8 (Y-X=15-7=8). Ainsi, une fois arrivé au couple (9;1), le couple formant la paire A=7 ce décalera de « -1 » [de (16;9) à (15;8)], et l'on recommencera à partir de (16;8) pour formé A=8. C'est comme si on passé de « 59 secondes et 0 minute » à «00 secondes et 1 minute », les secondes reviennent à leur valeur initiale. Sauf qu'ici, il n'y aura pas uniquement « seconde ; minute », mais « couple formant la paire A=8 ; couple formant la paire A=7 ; … ; couple formant la paire A=1 ».

→ De ce fait, le programme sera formé de 8 conditions imbriquées, et la boucle principale s'arrêtera quand le couple formant la paire A=1, sera écrit avec un X=1. « L'horloge » sera alors arrivé à la fin de son décompte, et aura faite toutes les combinaisons possibles, de couple de position (Y;X). De plus à chaque début de boucle, on fera un test pour identifier si c'est une suite de skolem. (En effet on identifiera une suite de Skolem, que lorsque chaque variables des couples formant les paire A=8 ; A=7 etc... sont différentes entre elles, donc un seul 1, un seul 2, …, un seul 16).

→ Pour réaliser le test sur les variables des huit couple (Y ;X), afin savoir si ils sont tous différent entre eux, et donc identifier que ce sont les bonnes positions pour former une suite de Skolem, on utilisera l'algèbre de Boole. En effet nous pourrions conjecturer « Si X1+X2+X3+...+X16=136, alors c'est une suite de skolem » (car 1+2+3+...+16=136). Cependant **cette conjecture est fausse**, car il y a d'autre moyen de former le chiffre 136 sans que pour autant que les 16 chiffres soit différents ! (exemple : 1+**2+2**+3+5+6+... +16=136, il y a 16 chiffres, et ne sont pas tous différents). Par contre avec l'algèbre de Boole, nous savons que 2^z, avec z un chiffre quelconque, donne un chiffre unique. Donc la somme de : 2^1+2^2+2^3+...+2^16= 131070 est **unique** ! De ce fait, nous ferons le test : **SI** 2^X1+2^X2+2^X3+...+2^X16, **vaut** 131070, alors nous identifions les couples de positions formant A=8,…, A=1 comme celle d'une suite de Skolem .

→ *remarque :* Le fait d'utiliser des couples de positons dans cette méthode de recherche successive pour placer les huit paires A, est qu'elle réduit **considérablement** le nombre de combinaisons possible à calculé. En effet si nous aurions utiliser le chiffre même, par exemple une incrémentation de +1 du chiffre 0000000000000001, il aurait fallut un temps démesuré (Mois ? Années?) pour trouver les 504 suites de skolem d'ordre huit. Alors qu'avec cette méthode, un PC de puissance standard (32 bits) ne met que 20 minutes seulement pour identifier les 504 suites. C'est pour cela que le support utilisé sera une programmation en C++, afin de bénéficier de la grande puissance et rapidité de calcule qu'offre un ordinateur.

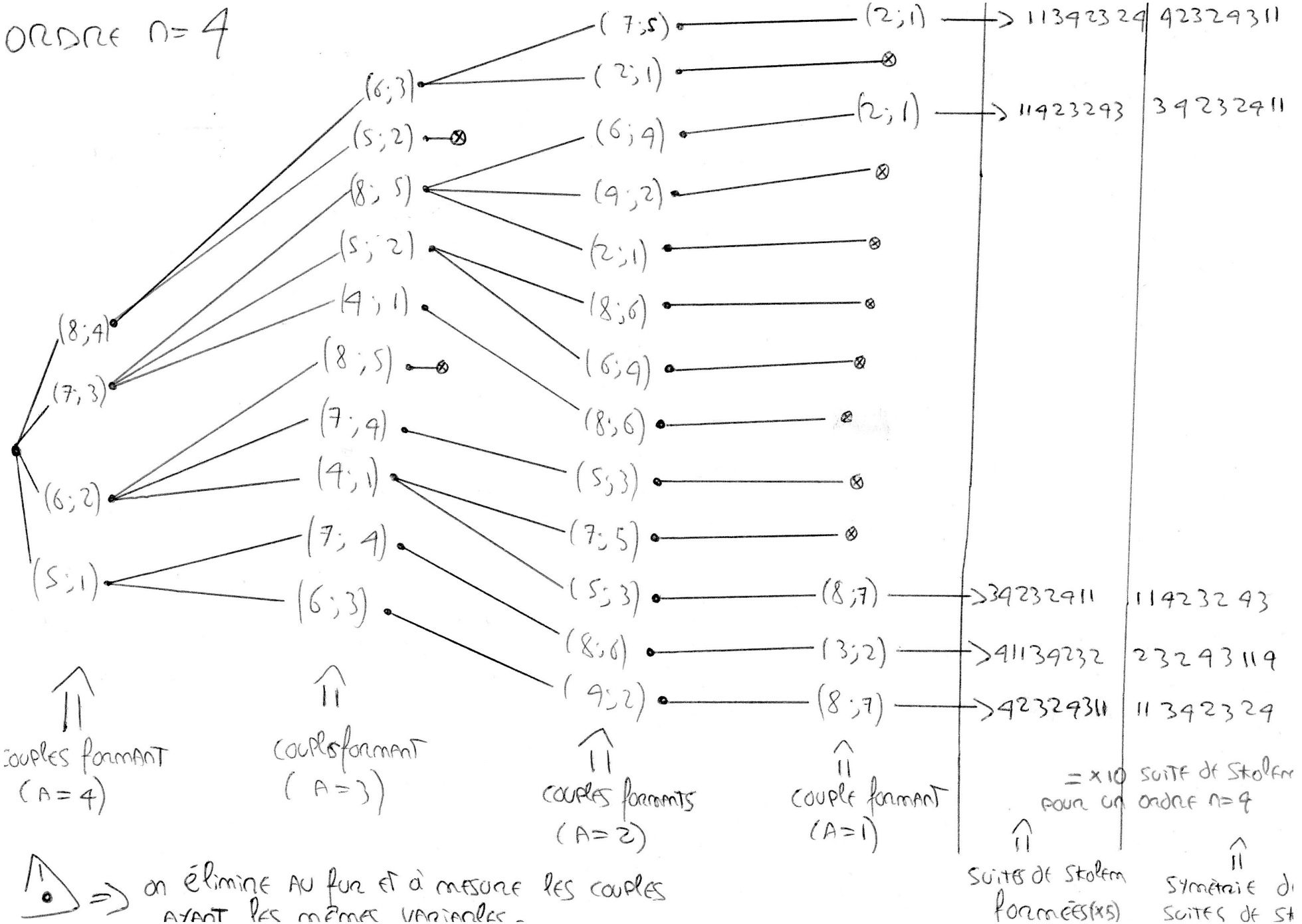
→ Représentation de la méthode sous forme d'arbre :

→ Le concept « d'horloge » utilisé précédemment pour expliqué la méthode de recherche successive de couple de position (Y;X), peut également ce visualiser sous la forme d'un arbre de solution . Page suivante, l'arbre illustrant la méthode expliqué précédemment, avec un ordre n=4 .

→ *ici le tableau de couples de position ayant permit la réalisation de l'arbre si-dessous :*

A =	Y-X						
1	(8-7)	(7-6)	(6-5)	(5-4)	(4-3)	(3-2)	(2-1)
2	(8-6)	(7-5)	(6-4)	(5-3)	(4-2)	(3-1)	
3	(8-5)	(7-4)	(6-3)	(5-2)	(2-1)		
4	(8-4)	(7-3)	(6-2)	(5-1)			

→ *remarque 1 :* Ici, l'ordre n=4, dans seulement 8 positions maximum
→ *remarque 2* : L'arbre représenté a parfois en bout de branche un cercle avec une croix dedans : cela est pour indiqué lorsque il n'y a plus de couple de position disponible pour lequel les variables le formant, posséderait des chiffres différents que les couples antérieurement utilisés dans la branche (car si des numéros de positions sont identiques, alors ce n'est pas une suite de skolem, donc une impasse). Cependant le programme fera quand même la suite des « branches », si les chiffres utiliser par les couples qu'il charge sont les mêmes (de part sa structure).
→ *remarque 3 :* Vous l'avez comprit, une fois les couples de positons (Y;X) identifiés, il suffira de mettre les différentes paires A à la place qu'indique leurs couples de positons respectif.


ORDRE n=4
(8;4)
(7;3)
(6;2)
(5;1)
(6;3)
(5;2)
(8;5)
(5;2)
(4;1)
(8;5)
(7;4)
(4;1)
(7;4)
(6;3)
(7;5)
(2;1)
(2;1)
(6;4)
(9;2)
(2;1)
(8;6)
(6;4)
(8;6)
(5;3)
(7;5)
(5;3)
(8;6)
(9;2)
(2;1)
(2;1)
(8;7)
(3;2)
(8;7)
11342324 42324311
11423243 34232411
34232411 11423243
41134232 23243114
42324311 11342324
couples formant (A=4)
couples formant (A=3)
couples formants (A=2)
couple formant (A=1)
Suites de Stolfm formées (x5)
= x10 suite de Stolfm pour un ordre n=9
Symétrie d
suites de St
=> on élimine au fur et à mesure les couples

→ *Approfondissement :* Si l'on voudrait améliorer le programme, il faudrait réussir à lui demander de sélectionner uniquement les couples qui ne contienne pas des chiffres déjà utilisées antérieurement (comme sur l'arbre fait ci-dessus) pour la formation d'une autre paire, ainsi, le programme fera une recherche extrêmement rapide, de l'ordre d'une 10ène de secondes. C'est à dire que, pour reprendre le concepts « d'horloge », au lieu de passer de « 57s et 08min » à « 58s et 8min » le programme demandera de passer de « 57s et 08min » à « 59s et 08min », donc sauté une seconde, de façon à ce que l'heure indiqué par l'horloge n'est pas de chiffre identique (en l'occurrence ici **8**).

→ Cahier des charges du programme en C++

(conditions de fonctionnement)

→ Synchroniser les 8 boucles entre elles, de manière à ce qu'elles ne soit pas en décalage l'une par rapport à l'autre.

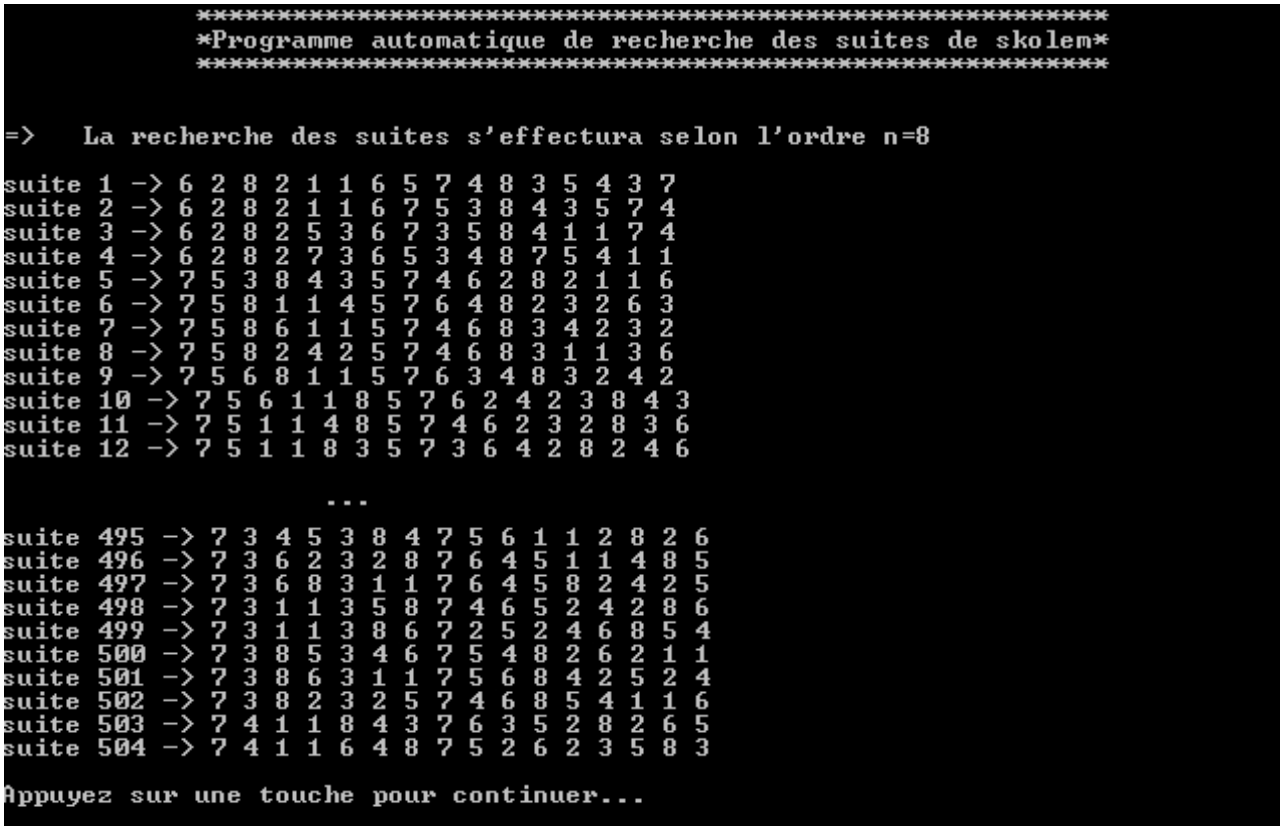
→ Permettre aux boucles de faire prendre successivement les valeurs des différents couple (Y;X) à deux variable unique. Une boucle pour chaque paire A à former.

→ Permettre au programme la reconnaissance d'une suite de Skolem par un test à chaque début de la boucle principale. Si le résultat est positif, faire afficher le résultat, sinon, laisser la boucle tourner encore.

→ Permettre au programme de réinitialiser le premier couple, quand la boucle est arriver au dernier. Comme par exemple (avec la paire A=8), quand (16;8) stocker dans les deux variables fixe de la boucle arrive au couple (9;1) par décrémentation, il faut a nouveau stocker le couple (16;8). (C'est comme remettre 00s quand on fait 59s 0 min +1s = 00s 1min).

→ permettre au programme d'afficher les 8 paires de la suite à la bonne place, une fois les 8 couples (Y;X) trouvés pour une suite de skolem. Soit, placé les différentes paires A sur un axe en fonction des positions des couples trouvé (pour un test déclaré positif pour la suite de Skolem par le programme).

→ Capture d'écran des 504 solutions trouvées par le programme en C++ :



→ Script Du programme en C++

(Avec explications des fonctions de programmation, la méthode étant décrite si-dessus)

```
#include<iostream>// Chargement de la bibliothéche faisant fonctionner l'affichage du programme
#include<stdio.h> // Ajoute des fonctions pour l'ajout de décimales
#include<cmath>// Ajoute certaines fonctions mathématique comme la racine carré et la puissance
// /!\ -> les accents ne sont pas permis en c++ sur la boite de dialogue du programme
using namespace std; // Espace de stockage du programme

int main()          // Début programme
{
    cout<<"             ************************************************************"<< endl;// En-tête
    cout<<"             *Programme automatique de recherche des suites de skolem*"<< endl;
    cout<<"             ************************************************************"<< endl;
    cout<<endl<<endl; // Retour à la ligne
    cout<<"=>   La recherche des suites s'effectura selon l'ordre n=8"<<endl;

    double a=16, b=8, c=16, d=9, e=16, f=10, g=16, h=11, i=16,
           j=12, k=16, l=13, m=16, n=14, o=16, p=15, q=0, r=0,
           X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12,
           X13, X14, X15, X16; //Annonce toutes les variables que l'on utilisera ensuite.

while(p!=0)                     // La boucle s'arréte pas, tant que p est différent de 0...
{                          // ...car une fois p à 0, toutes les combinaisons ce serons effectuées.

   if((pow(2,a)+pow(2,b)+pow(2,c)+pow(2,d)+pow(2,e)+pow(2,f)+pow(2,g)+pow(2,h)+pow(2,i)+pow(2,j)+
       pow(2,k)+pow(2,l)+pow(2,m)+pow(2,n)+pow(2,o)+pow(2,p))==131070) //Test pour savoir si...
    {   //...la combinaison des variables des couples de position trouvées sont bien toutes ...
        //...différentes entre elle = Suite de skolem = Afficher la suite (Cf étude théorique).

 q++; // Incrémentation de la variable comptant le nombre de suite trouvée (504 suites trouvées Max)

 X1=a-b; // toutes les positions X"chiffre" (les paires A) sont affectées de la valeur ...
 X2=a-b; // ... de la différences Y-X de leur couple associé.
 X3=c-d;
 X4=c-d;
 X5=e-f;
 X6=e-f;
 X7=g-h;
 X8=g-h;
 X9=i-j;
 X10=i-j;
 X11=k-l;
 X12=k-l;
 X13=m-n;
 X14=m-n;
 X15=o-p;
 X16=o-p;
```

```
//Puis on affiche les différentes paires A, en fonction de leurs couples de positions...
//...à l'aide de deux tableaux de char (cf si-dessous).
int valeur[16] = {X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13,X14,X15,X16}; //Valeurs à afficher.
//               |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
int ordre[16] =  {a, b, c, d ,e, f, g,  h, i ,j,  k,  l  ,m  ,n,  o,  p};  //Ordre de ces valeurs.
cout<<endl<<endl; //retour à la ligne.
cout<<"Suite "<<q<<"-> "; //Affichage du compteur de suite.
for (int r=0; r<16; r++) // Boucle affichant caractères par caractère les valeurs par ordre.
{cout << valeur[ordre[r]];}
cout<<endl;//retour à la ligne.

   }

  a--;                  // Ici ce déroule l'étape de "l'horloge" à 8 composantes, je rappel que...
  b--;                  //... le principe est expliqué dans l'étude théorique du DM, nous ne ...
       if(b==0)         //...détaillerons donc pas une deuxième fois la méthode. Je précise juste...
       {a=16;           //... que par exemple "a--" décrémente la variable "a" de "1".
        b=8;
        c--;
        d--;
            if(d==0)
            {c=16;
             d=9;
             e--;
             f--;
                 if(f==0)
                 {e=16;
                  f=10;
                  g--;
                  h--;
                        if(h==0)
                        {g=16;
                         h=11;
                         i--;
                         j--;
                               if(j==0)
                               {i=16;
                                j=12;
                                k--;
                                l--;
                                      if(l==0)
                                      {k=16;
                                       l=13;
                                       m--;
                                       n--;
                                            if(n==0)
                                            {m=16;
                                             n=14;
                                             o--;
                                             p--;
                                            }
                                      }
                               }
                        }
                 }
            }
       }
}


    system("pause"); // Mise en pause de l'affichage à la fin du prog. pour pouvoir...
                     //...lire et exploiter les résultats trouvés.
    return 0;     // Marque la fin du programme, ferme le "int main()" du départ.
}
```