حساب التباديل بالطريقة النسبية

م.م. رافد فياض حمدي*

المستخلص:

إن استخراج التباديل بالطريقة التقليدية وباستخدام الحاسبة لمجموعة مكونة من **n** عنصر تعتمد على تحويل الأرقام بالنظام العشري من **1** إلى n^n-1 لنظام الأرقام النوني (للأساس **n**) أي جميع المراتب تأخذ الأرقام **0** ، **1** ، **2** ، ... ، **n - 1** ويتكون الرقم من **n** مرتبة . عند كل زيادة لرقم واحد نجري عملية مقارنة بين أرقام المراتب فإذا كانت جميع الأرقام مختلفة يتم الأخذ بهذا الرقم على انهُ عنصر من عناصر التباديل وبخلاف ذلك تكون المقارنة فاشلة وهذا يعني وجوب تحويل كامل الرقم من النظام العشري إلى النظام النوني وبناءً على ذلك سوف يتم استخدام الجملة الشرطية $n^{n+1}(n-1)/2$ مرة . كذلك لحساب إشارة التباديل يتطلب نفس العدد من الجمل الشرطية مما يضاعف زمن تنفيذ البرنامج .

أما بطريقة البحث نستطيع استخراج التباديل لنفس المجموعة بزمن يصل إلى **2 %o** عندما $n\leq 10$ من زمن البرنامج التقليدي وتقل هذهِ النسبة بزيادة **n** وذلك لقلة استخدام الجمل الشرطية لتصل إلى $(e-1)n!$ مرة منها $(e-2)n!$ مرة تكون المقارنات فاشلة . والطريقة تتلخص بتحويل الأرقام العشرية من **1** إلى **n ! - 1** من النظام العشري إلى نظام المفكوك حيث تكون المرتبة الأولى للأساس **2** و المرتبة الثانية للأساس **3** وصولاً لأخر مرتبة للأساس **n** وهذا الرقم يتكون من **n - 1** مرتبة . وهذهِ الطريقة لا تتطلب تحويل الرقم بشكل كامل إلى نظام المفكوك بل تتوقف عملية التحويل عند ظهور أول رقم اكبر من الصفر في المراتب الدنيا للرقم بنظام المفكوك ويعتبر أساس هذهِ المرتبة وهو **X** الذي يرمز إلى رقم الموقع الذي ستبدأ فيهِ عملية الاستبدال . حيث سيتم استبدال العنصر بالموقع **X** مع العنصر بالموقع الأول ثم استبدال العنصر بالموقع **X - 2** مع العنصر بالموقع الأول ثم استبدال العنصر بالموقع **X - 4** مع العنصر بالموقع الأول وصولاً لأدنى عنصر وان هذهِ الطرية لا تتطلب المقارنة بين عناصر المجموعة أو أي عملية مقارنة أخرى . وتسمى هذهِ التباديل بالتباديل النسبية لان عملية الاستبدال تجري على العنصر السابق وليس على العنصر الافتراضي الذي يكون فيهِ الترتيب بشكل متصاعد . أما حساب الإشارة تتم بمعادلة بسيطة وهي ضرب إشارة الترتيب السابق بـ $(-1)^g$ و **g** هي عدد حالات الاستبدال التي جرت للحصول على الترتيب الحالي.

Abstract:

To extraction the Permutations by used computer there is a classical method it is recap if we have set of n elements. it's rely upon conversion the numbers from 1 to n^n-1 in denary system (decimal base 10) to the system numbers on base n namely whole orders get the subsequent integer numbers 0 ,1 , 2 , 3 , … , n – 1 and the n base number contain n orders . at each increasing of one we must compare among all numbers in orders if each number different from all other numbers we consider this number as element of Permutations otherwise the comparisons fruitless it's mean we must convert entire number from denary system to n base system construction on this will using IF Statements $n^{n+1}(n-1)/2$ times such that for calculation signal of each rearrangement of

* مدرس مساعد/الجامعة المستنصرية/كلية الإدارة والإقتصاد/قسم الإحصاء

مقبول للنشر بتاريخ **2011/3/3**

Permutations we employ the same figure of IF Statements this operation will duplicate the time that spend to program's execution .

The program depends on method in this paper extraction the Permutations in(2%) when $n \leq 10$ from the time that spends in classical program. This percentage decrease when n increase because reduce to employ IF Statements it is approached (e – 1) n! time. from this figure (e – 2) n! time fruitless of using IF Statement. This method brief by covert denary system numbers from 1 to n! – 1 to factorial system number and this idea is essential of method. first order of factorial number on base 2 , second on base 3 , until last order on base n and factorial number content n – 1 orders . in this method didn't need to convert unmitigated number but we stopping at appearance first number greater than zero in lower position the base of this order is X in the same time X consider site that swap begin . the process start by change the element in location X with the element in location 1 and change element in location X – 2 with the element in location 1 so on continuance until inferior location grater than 1 this replacement is major think of the method . The Permutations called relative Permutations since replacement performs on anterior rearrangement not on elementary rearrangement. To account signal of present rearrangement product signal of foregoing rearrangement with -1 power figure replacement in current rearrangement. We see did not construct any comparison among elements them self.

1 ـ المقدمة Introduction

اهتم علماء الرياضيات Mathematician منذ القدم بإيجاد جذور المعادلات Roots of equation وكان للبابليين الدور الرئيسي في هذا المجال بتمكنهم من إيجاد الجذور للمعادلات من الدرجة الثانية Quadratic والتي تعرف بطريقة الدستور حتى بداية القرن السادس عشر الميلادي تمكن الايطاليون من التوصل إلى طريقة لإيجاد جذور المعادلات من الدرجة الثالثة Cubic والرابعة Quartic وخلال ثلاثة قرون التالية حاول المهتمين بالرياضيات وضع صيغة عامة لجذور المعادلات من الدرجة الخامسة فأكثر والتي تعرف بالمتعدد الحدود Polynomial .

في عام 1815 تمكن A.L. Cauchy من توضيح فكرة التباديل Permutations والمجموعات المتماثلة Symmetric groups وقسم التباديل إلى دورات منفصلة Disjoint cycle وقد ساعدت التباديل في إيجاد جذور المعادلات من الدرجة الخامسة فأكثر .

في العام 1824 اثبت النرويجي Niels Abel (1829 – 1802) بعدم وجود صيغة عامة لحل المعادلات من الدرجة الخامسة فأكثر وقد استند في ذلك على ما توصل أليه J . L . Lagrange (1770) .

ثم استطاع لاحقاً الرياضي الفرنسي Evariste Galois (1832 – 1811) والذي قتل قبل بلوغه إحدى وعشرون سنة من العمر إن يبرهن جميع النظريات المتعلقة بالتباديل Permutations ولحد ألان تعتمد المصادر الحديثة على تلك البراهين .

لمصطلح التباديل Permutations معنيين الأول وهو تباديل n من n ونرمز لهُ $_nP_n = n!$ ويعتبر مرادف للتراتيب Rearrangement وهذا المعنى سوف نستخدمهُ في البحث انسجاماً مع المصادر المأخوذ منها الجانب النظري حيث تُستخدم التباديل بدل التراتيب ومن الواضح إن التباديل في المعنى الأول هو حالة خاصة من المعنى الثاني وهو المعنى العام فالتباديل r من n ونرمز لهُ بالصيغة التالية : $_nP_r = n! / (n-r)!$ وهو حساب عدد الطرق التي يمكن أن نختار فيها عينة بحجم r عنصر من مجموعة مكونة من n عنصر مع أخذ تراتيب العينة بنظر الاعتبار وتعتبر عملية سحب بدون إرجاع Without replacement . من الطبيعي أن عدد طرق التباديل تعتبر من البديهيات لذلك فالبحث يهتم بترتيب العناصر بتلك الـ n! طريقة ولتجنب اللبس سوف نقدم الجانب النظري قبل بقية الفقرات لتوضيح المقصود بالتباديل .

2 ـ الجانب النظري The Theoretical pane

تعرف التباديل Permutations لمجموعة Set مكونة من n عنصر Elements على أنها دالة متقابلة Bijection على نفسها that is one – to – one and onto f : ⎕ ζ ⎕ والشكل الرياضي للدالة

$$f = \begin{pmatrix} 1 & 2 & 3 & & n \\ f(1) & f(2) & f(3) & \cdots & f(n) \end{pmatrix}$$

حيث تسمى تلك المجموعتان بالمجموعات المتماثلة Symmetric groups لاحتوائها على نفس العدد من العناصر ولكن ترتيب Rearrangement العناصر في المجموعة الثانية قد يختلف عن الأولى .
وعلى ضوء هذا التعريف سنعبر عن المجموعة S_2 المتكونة من عنصرين بالشكل التالي :

أما المجموعة S_3 فتتكون من 6 = !3 عناصر وبالشكل التالي :

$$S_2 = \left\{ \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} , \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \right\}$$

$$S_3 = \left\{ \begin{pmatrix} 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix} , \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} , \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} , \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} , \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} , \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \right\}$$

من خلال المثال التالي سوف نوضح بعض المفاهيم المتعلقة بالتباديل . حيث عندما n = 5 سوف يتم استبدال العنصر بالموقع الأول مع العنصر بالموقع الثالث ومن ثم يتم استبدال العنصر بالموقع الثالث مع العنصر بالموقع الرابع هذا من جهة ومن جهة أخرى قد تم استبدال عناصر الموقعين الثاني والخامس احدهما بالأخر ويعبر عنها رياضياً بالصيغة التالية :

لكون الدورتين $\begin{pmatrix} 1 & 3 & 4 \end{pmatrix}$ و $\begin{pmatrix} 2 & 5 \end{pmatrix}$ منفصلتين Disjoint فمن الممكن التقديم والتأخير كما موضح أعلاه وتسمى الدورة $\begin{pmatrix} 2 & 5 \end{pmatrix}$ بالدورة الثنائية 2 – cycle لكون قد تم استبدال عنصرين فقط أما الدورة

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 3 & 4 \end{pmatrix}\begin{pmatrix} 2 & 5 \end{pmatrix} = \begin{pmatrix} 2 & 5 \end{pmatrix}\begin{pmatrix} 1 & 3 & 4 \end{pmatrix}$$

$\begin{pmatrix} 1 & 3 & 4 \end{pmatrix}$ فتسمى بالدورة الثلاثية 3 – cycle لأنهُ قد تم تغيير مواقع ثلاثة عناصر وفي هذا المثال أولاً تم

استبدال العنصرين بالموقعين الأول والثالث ومن ثم تم استبدال العنصرين بالموقعين الثالث والرابع مع بعضهما وبالتالي يمكن كتابة عملية الاستبدال هذهِ بالصيغة التالية :

$$\begin{pmatrix} 1 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 3 \end{pmatrix} \begin{pmatrix} 3 & 4 \end{pmatrix}$$

ولا يمكن كتابتها بالعكس لكون الدورتين الثنائيتين **2 – cycle** $\begin{pmatrix} 1 & 3 \end{pmatrix}$ و $\begin{pmatrix} 3 & 4 \end{pmatrix}$ **non – disjoint** أي غير منفصلتين لان الموقع الثالث متكرر في كلا الدورتين كما مبين بالصيغة التالية :

ومن الممكن إعادة كتابة الصيغة أعلاه بالشكل التالي :

$$\begin{pmatrix} 1 & 3 & 4 \end{pmatrix} \neq \begin{pmatrix} 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 2 & 1 & 3 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 3 \end{pmatrix} \begin{pmatrix} 3 & 4 \end{pmatrix} \begin{pmatrix} 2 \end{pmatrix} \begin{pmatrix} 5 \end{pmatrix}$$

حيث في العادة يتم إهمال الدورة **1 – cycle** مثل $\begin{pmatrix} 5 \end{pmatrix}$, $\begin{pmatrix} 2 \end{pmatrix}$ لعدم حصول عملية استبدال فيها .

الأمثلة التالية توضح عملية ضرب الدورات

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 2 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 4 & 5 \end{pmatrix}$$

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 3 & 4 & 5 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 6 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 6 & 2 & 3 & 4 & 5 \end{pmatrix}$$

تنقسم التباديل **Permutations** أي **n!** ترتيب إلى عدة مجاميع تبعاً لعدد العناصر التي سيتم استبدالها فيوجد $\frac{n(n-1)}{2}$ ترتيب ناتج من استبدال عنصرين فقط احدهما بالأخر وتسمى دورة ثنائية **2 – cycle** بينما يوجد $\frac{n(n-1)(n-2)}{3}$ ترتيب ناتج من استبدال ثلاثة عناصر وتسمى دورة ثلاثية **3 – cycle** وهكذا وصولاً إلى **r – cycle** والصيغة العامة لها هي $\frac{n!}{(n-r)!} \times \frac{1}{r}$ أو $_nP_r \times \frac{1}{r}$ وإذا احتوى الاستبدال أكثر من دورة منفصلة **Disjoint** مثلاً الدورة **r_1 – cycle** والدورة **r_2 - cycle** وكان $r_1 \neq r_2$ فيوجد $\frac{n!}{(n-r_1)!} \times \frac{1}{r_1} \times \frac{(n-r_1)!}{(n-r_1-r_2)!} \times \frac{1}{r_2}$ ترتيب ويمكن تبسيط الصيغة على النحو التالي

$\frac{n!}{(n-r_1-r_2)!} \times \frac{1}{r_1} \times \frac{1}{r_2}$ كذلك أذا احتوى الاستبدال على ثلاث دورات منفصلة بحيث $r_1 \neq r_2 \neq r_3$ فيوجد $\frac{n!}{(n-r_1-r_2-r_3)!} \times \frac{1}{r_1} \times \frac{1}{r_2} \times \frac{1}{r_3}$ ترتيب وهكذا وصولاً إلى **k** من الدورات المنفصلة بحيث $r_1 \neq r_2 \neq \ldots \neq r_k$ فيوجد $\frac{n!}{(n-r_1-r_2-\ldots-r_k)!} \times \frac{1}{r_1} \times \frac{1}{r_2} \times \ldots \times \frac{1}{r_k}$ ترتيب إما في حال تساوي r_2

و r_1 فنقسم عدد التراتيب على !2 وإذا كانت ثلاث دورات متساوية $r_1 = r_2 = r_3$ فنقسم عدد التراتيب على !3 وهكذا وصولاً إلى $r_1 = r_2 = \dots = r_k$ فيتم قسمة التراتيب على !k .

الجداول التالية توضح ما شرحناهُ سابقاً . العمود الأول في الجداول Cycle Structure يحدد عدد العناصر التي سيتم استبدالها مثلاً (2 1) لا تعني استبدال العنصرين في الموقع الأول والثاني مع بعضهما وإنما استبدال أي عنصرين مع بعضهما أي دورة ثنائية cycle – 2 و (3 2 1) تعني cycle – 3 وهكذا وصولاً إلى (r ... 2 1) وتعني r – cycle استبدال r عنصر مع بعضهم البعض أما العمود الثاني Number فيحسب عدد التراتيب الناتجة من r استبدال ويجب أن يكون المجموع الكلي للتراتيب !n . أما العمود الثالث فيحدد الإشارة Signal وتكون الإشارة موجبة أذا كانت عدد العناصر المستبدلة فردية وسالبة للدورات الزوجية أما في حال احتواء العمود الأول على أكثر من دورة منفصلة Disjoint عندها نضرب إشارة الدورات مع بعضها للحصول على الإشارة النهائية .

الجدول الأول عندما n = 4 وفيهِ 24 تبديل والثاني عندما n = 5 وفيهِ 120 تبديل والثالث عندما n = 6 وفيهِ 720 تبديل .

الجدول رقم (1) عندما n = 4

Cycle Structure	Number	Signal
(1)	1	+
(1 2)	6	–
(1 2 3)	8	+
(1 2 3 4)	6	–
(1 2) (3 4)	3	+
	24	

الجدول رقم (2) عندما n = 5

Cycle Structure	Number	Signal
(1)	1	+
(1 2)	10	–
(1 2 3)	20	+
(1 2 3 4)	30	–
(1 2 3 4 5)	24	+
(1 2) (3 4 5)	20	–
(1 2) (3 4)	15	+
	120	

جدول رقم (3) عندما n = 6

Cycle Structure	Number	Signal
(1)	1	+
(1 2)	15	–
(1 2 3)	40	+
(1 2 3 4)	90	–
(1 2 3 4 5)	144	+
(1 2 3 4 5 6)	120	–
(1 2) (3 4 5 6)	90	+
(1 2) (3 4 5)	120	–
(1 2) (3 4)	45	+
(1 2) (3 4) (5 6)	15	–
(1 2 3) (4 5 6)	40	+
	720	

لاستكمال تعريف التباديل فان كل عنصر في موقع معين لابد من ظهورهُ في نفس الموقع أو في موقع أخر بعد سلسلة من عمليات الاستبدال ومن الملاحظ في الجداول السابقة أن جميع عمليات الاستبدال تتم على الترتيب الأساسي للعناصر وهو $\begin{pmatrix} 1 & 2 & 3 & \dots & n \end{pmatrix}$ وتسمى هذهِ العملية بالتباديل المطلقة **Absolute** ولكون هذا النوع من التباديل يستغرق الكثير من الوقت عند برمجته على الحاسبة لذلك استخدمنا في هذا البحث التباديل النسبية **Relative** أي أن كل استبدال يعتمد على الذي قبلهُ وسوف نشرح هذهِ الطريقة في الجانب التطبيقي . في منتصف القرن العشرين استخدمت الحاسبات في المجالات العلمية بشكل عام والرياضيات بشكل خاص ولكن في الربع الأخير من القرن الماضي حدث تطور جوهري في صناعة الحاسبات ولا يزال هذا التطور مستمر لحد ألان حيث كانت سرعة المعالج **CPU** من النوع **Intel 8086** المستخدم في الحاسبات الشخصية **Personal Computer** **3.6 MHz** والمواصفات الأخرى لهذا الجيل كسعة الذاكرة **RAM** لا تتعدى **64KB** وكذلك السعة التخزينية للقرص الصلب **Hard Disk** لا تتعدى **5 MB** إما مواصفات الجيل الحالي للحاسبات كالتي تعمل بالمعالج **Pentium 4** والتي تصل سرعتهُ إلى **3200 MHz** أو (**3.2 GHz**) كذلك تطور بقية المواصفات كسعة الذاكرة والتي تصل إلى **2GB** وسعة القرص الصلب لتصل إلى **200 GB** كذلك أضيفت تقنيات متعددة للتخلص من وجود خطوة تحد من سرعة الحاسبة فأصبحت سرعة انتقال البيانات إلى الذاكرة **3200 MB / s** وتم إضافة ذاكرة تسمى **Cache** والتي تقترب سرعتها من سرعة المعالج كي تقلل من فترة بقاء المعالج دون عمل **Idle** بانتظار استلام أو إرسال البيانات إلى المكونات الأخرى للحاسبة وبالنتيجة أصبح البرنامج الذي يستغرق عشر ساعات في الحاسبات القديمة مثل **MSX200** والتي تعمل بمعالج **3.6 MHz** لا يستغرق سوى دقيقة واحدة عند تنفيذهُ على حاسبات تعمل بمعالج **3200 MHz Pentium 4** مما دفع الرياضيون **Mathematicians** إلى برمجة أعمالهم على الحاسبة ولكن بالرغم من ذلك فان التباديل **Permutations** لا تزال صعبة الاستخدام بحيث أن البرنامج الذي يحسب قيمة المحدد **Determinant** اعتماداً على التباديل يستغرق عدة ساعات أذا زاد حجم المحدد عن **n = 10** وذلك لان البرنامج التقليدي **Classical Program** يحتاج إلى استخدام جمل المقارنة (أو الجمل الشرطية **IF Statements**) $\frac{n^{n+1}(n-1)}{2}$ مرة وهذهِ الجمل تتطلب الكثير من الوقت لانجازها وكلما قل استخدام هذهِ الجمل زاد من سرعة وكفاءة البرنامج كذلك الجمل الحلقية تحتاج الكثير من الوقت لاحتوائها ضمنياً على الجملة الشرطية مثل الجمل الحلقية التالية :

(FOR … NEXT) & (DO … LOOP)

لذلك تم الاتجاه إلى التقنيات البديلة لتجنب استخدام التباديل فعلى سبيل المثال فقد صمم **E.Miller** في العام 1978 برنامج قصير بلغة **FORTRAN IV** لحساب قيمة المحدد بطريقة **Expanding by Columns or Rows** ولحد ألان يستخدم هذا البرنامج وأصبح إحدى دوال برنامج **Microsoft Excel** .

نلاحظ في الجدول رقم (4) العمود الأول ويبين عدد عناصر المجموعة بينما يحسب العمود الثاني عدد التباديل أي عدد حالات السحب بدون إرجاع **Without Replacement** وهي تمثل عدد مرات استخدام الجمل الشرطية الناجحة بينما العمود الثالث يحسب عدد حالات السحب بالإرجاع **With Replacement** وعددها n^n مضروبة في عدد المقارنات لكل ترتيب وهو $\frac{n(n-1)}{2}$ والناتج يساوي $\frac{n^{n+1}(n-1)}{2}$ وهو يمثل عدد مرات استخدام الجمل الشرطية الكلية أما العمود الرابع والذي يشير إلى نسبة استخدام الجمل الشرطية الناجحة إلى المجموع الكلي للجمل الشرطية ويساوي $\frac{n!}{\frac{n^{n+1}(n-1)}{2}}$ أو $\frac{2\,n!}{n^{n+1}(n-1)}$

وأخيراً العمود الخامس هو مقلوب العمود الرابع .

الجدول رقم (4)

n	n!	$\frac{n^{n+1}(n-1)}{2}$	$\frac{2\,n!}{n^{n+1}(n-1)}$	$\frac{n^{n+1}(n-1)}{2n!}$
4	24	1536	0.01562500	64
5	120	31250	0.00384000	260.4167
6	720	699840	0.00102881	972
7	5040	17294403	0.00029142	3431.429
8	40320	469762048	0.00008583	11650.84
9	362880	13947137604	0.00002602	38434.57
10	3628800	450000000000	0.00000806	124007.9

نلاحظ أن البيانات في العمود الرابع تأخذ شكل الدالة الآسية التالية :

$$y = 0.0483\, e^{-1.2564x} \qquad 4 \le x \le 10 \qquad .... \quad (1)$$

أما البيانات في العمود الخامس فهي أيضاً دالة آسية وبالصيغة التالية :

$$y = 20.694\, e^{1.2564x} \qquad 4 \le x \le 10 \qquad .... \quad (2)$$

ومن خلال المعادلتين (1) و (2) نلاحظ كلما زادت قيمة n أدى ذلك إلى زيادة آسية في عدد الجمل الشرطية الفاشلة وبالتالي هدر في وقت تنفيذ البرنامج وعدم الجدوى من استخدام البرنامج التقليدي للتباديل .

3 – 2 هدف البحث The Goal

أن الهدف الأساسي للبحث هو الوصول إلى طريقة فعالة في استخراج التباديل باستخدام الحاسبة وتنسجم مع لغات البرمجة بحيث انخفض عدد جمل المقارنة (أو الجمل الشرطية) إلى ! 1.72 n منها !n جملة ناجحة وعدد الجمل الفاشلة هو ! 0.72 n جملة والجدول رقم (5) وفيهِ العمود الأول ويبين عدد عناصر المجموعة بينما يحسب العمود الثاني عدد الجمل الشرطية الناجحة والتي تساوي عدد التباديل ناقص واحد لاعتماد البحث على التباديل النسبية Relative والترتيب الأساسي لا يدخل في عملية الاستبدال والعمود الثالث يحسب عدد الجمل الشرطية الفاشلة والعمود الرابع يحسب مجموع الجمل الشرطية الناجحة والفاشلة بينما يحسب العمود الخامس والذي يشير إلى نسبة استخدام الجمل الشرطية الناجحة إلى المجموع الكلي للجمل الشرطية وأخيراً العمود السادس هو مقلوب العمود الخامس.

الجدول رقم (5)

n	n ! -1	False	Summation	ratio	ratio inverse
4	23	14	37	0.621622	1.608695652
5	119	82	201	0.59204	1.68907563
6	719	512	1231	0.584078	1.712100139
7	5039	3614	8653	0.582341384	1.717205795
8	40319	28954	69273	0.582030517	1.718122969
9	362879	260642	623521	0.581983606	1.718261459
10	3628799	2606492	6235291	0.581977489	1.718279519

نلاحظ في هذا الجدول بيانات العمود الخامس تكون شبه ثابتة أي $y = 0.58 \quad for \;\; all \;\; x$

بالمقارنة مع بيانات العمود الرابع في الجدول رقم (4) والتي تأخذ شكل الدالة الآسية رقم (1) كما ونلاحظ بيانات العمود السادس في الجدول رقم (5) تكون شبه ثابتة أي $y=1.72 \quad for \ all \ x$ بالمقارنة مع بيانات العمود الخامس في الجدول رقم (4) والتي تأخذ شكل الدالة الآسية رقم (2) ونتيجةً لانخفاض الجمل الشرطية فقد انخفض زمن تنفيذ البرنامج بالمقارنة مع البرنامج التقليدي للتباديل هذا بالإضافة إلى انخفاض الجمل الحلقية لتكن جملتين فقط بطريقة البحث بالمقارنة مع n+2 جملة حلقية للبرنامج التقليدي والجدول رقم (6) يعكس نسبة زمن التنفيذ بين الطريقتين اعتماداً على عدد الجمل الشرطية المستخدمة في كليهما فالصف الثاني مأخوذ من الجدول رقم (5) وهو العمود الرابع . والصف الثالث مأخوذ من الجدول رقم (4) وهو العمود الثالث أما الصف الرابع في الجدول رقم (6) فهو نسبة

الجدول رقم (6)

n	4	5	6	7	8	9	10
Summation	37	201	1231	8653	69273	623521	6235291
$\frac{n^{n+1}(n-1)}{2}$	1536	31250	699840	17294403	469762048	13947137604	450000000000
Ratio	41.51	155.47	568.51	1998.66	6781.32	22368.35	72169.85

القيم في الصف الثالث إلى الصف الثاني ومن الجدول نستنتج عندما n = 7 يكون البرنامج تقريباً أسرع بـ 2000 مرة من البرنامج التقليدي وهكذا لبقية قيم الصف الرابع هذا من الناحية النظرية أما عملياً فقد تم تنفيذ كلا البرنامجين على نفس الحاسبة واستغرق البرنامج التقليدي 17 دقيقة بينما استغرق برنامج البحث اقل من ثانيتين أي أسرع بـ 500 مرة . وسبب الفرق بين السرعتين النظرية و التطبيقية هو احتواء برنامج البحث على جمل حسابية أكثر من البرنامج التقليدي مما يتطلب نقل بيانات من والى الذاكرة RAM .

4 – الجانب التطبيقي The Application

من أهم مميزات هذهِ الطريقة هو تجنب المقارنة بين العناصر ذاتها سواء لاستخراج التباديل أو لمعرفة أشارة كل تبديل بينما الطريقة التقليدية تعتمد على المقارنة بين العناصر سواء لاستخراج التباديل أو للإشارة . حيث تعتمد طريقة البحث على إنشاء عداد خاص تكون المرتبة الأولى فيهِ بنظام الأعداد الثنائي و تأخذ القيم إما 0 أو 1 والمرتبة الثانية بنظام الأعداد الثلاثي و تأخذ القيم أما 0 أو 1 أو 2 والمرتبة الثالثة بنظام الأعداد الرباعي و تأخذ القيم أما 0 أو 1 أو 2 أو 3 وصولاً إلى أخر مرتبة بنظام الأعداد النوني و تأخذ القيم أما 0 أو 1 أو 2 أو 3 أو 4 أو ... أو n – 1 .تعتمد عدد المراتب على عدد عناصر المجموعة المراد أجراء الاستبدال لها فان كانت المجموعة متكونة من n عنصر فعدد المراتب تكون n – 1 مرتبة . يسمى هذا الرقم Factorial Number الرقم بصيغة المفكوك . من خلال الجملة الحلقية سوف نعطي تسلسل لكل ترتيب يبدأ بالرقم 0 وينتهي بالرقم n! – 1 وهذهِ الأرقام بالنظام العشري حيث تعتمد عملية استبدال العناصر على تحويل رقم التسلسل من النظام العشري إلى النظام بصيغة المفكوك فأول مرتبة تحتوي على رقم اكبر من 0 سوف تحدد موقع العنصر الرئيسي الذي ستبدأ بهِ عملية الاستبدال من خلال العلاقة التالية :

موقع العنصر الرئيسي الذي ستبدأ بهِ عملية الاستبدال = أصغر مرتبة بصيغة المفكوك لا تساوي صفر + 1 و يكون التسلسل 0 للترتيب الأساسي والتسلسل رقم 1 لأول ترتيب سيستبدل العنصر بالموقع الثاني مع العنصر بالموقع الأول حيث ستجري عملية الاستبدال على الترتيب السابق وفي هذهِ الحالة الترتيب الأساسي ثم نحول رقم 2 إلى الرقم بصيغة المفكوك ونحدد أول مرتبة تحتوي على رقم اكبر من 0 ونطبق العلاقة أعلاه لتحديد الموقع الذي سوف تتم بهِ عملية الاستبدال على الترتيب السابق أي الترتيب الذي يحمل التسلسل 1 وليس على الترتيب الأساسي وتسمى هذهِ الطريقة بالتباديل النسبية Relative مثلاً للحصول على الترتيب الذي يحمل التسلسل I نحول الرقم I من الصيغة العشرية (النظام العشري) إلى صيغة المفكوك ونحدد اصغر مرتبة لا تحتوي على صفر ونضيف عليها واحد ليكون الرقم الناتج هو موقع العنصر الذي ستبدأ بهِ عملية الاستبدال والذي سنجريهِ على الترتيب الذي يحمل التسلسل I – 1 ولذلك عدد عناصر مجموعة التباديل النسبية تكون n! – 1 بدل n! في التباديل المطلقة والموضحة في الجانب النظري . سنعبر عن المجموعة S_2 النسبية المتكونة من عنصر واحد بالشكل التالي :

$$S_2 = \left\{ \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \right\}$$

أما المجموعة S_3 النسبية فتتكون من $3! - 1 = 5$ عناصر وبالشكل التالي :

$$S_3 = \left\{ \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 3 & 1 & 2 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 3 & 2 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 3 & 1 \\ 3 & 2 & 1 \end{pmatrix} \right\}$$

الترتيب الأساسي يعتبر الترتيب الابتدائي أو الترتيب المستقل أو الافتراضي الذي سوف ننطلق منهُ للحصول على الترتيب التالي ولذلك لا يدخل في المجموعة كعنصر لذلك تنقص عناصر التباديل بواحد أي $n! - 1$ عنصر . يعتبر تحويل التسلسل من النظام العشري إلى صيغة المفكوك (أو نظام المفكوك) من الأمور الجوهرية في هذا البحث ولأجراء التحويل يوجد عدة طرق كحال تحويل الرقم من النظام العشري إلى الثنائي أو الثماني أو الستة عشري وأسهل طريقة قابلة للبرمجة لتحويل الرقم من النظام العشري الى نظام المفكوك هي قسمة الرقم العشري على 2 ووضع المتبقي من القسمة بالمرتبة الأولى لنظام المفكوك أما ناتج القسمة فيقسم على 3 ووضع المتبقي من القسمة في المرتبة الثانية لنظام المفكوك إما ناتج القسمة فيقسم على 4 ونستمر بهذهِ الإلية وصولاً لأخر مرتبة علماً أن عدد المراتب لنظام المفكوك تقل عن عدد عناصر المجموعة المراد استخراج التباديل لها بواحد أي أذا كانت المجموعة متكونة من n عنصر فأن عدد المراتب تكون n - 1 مرتبة وسبب طرح الواحد من n أيضاً لنفس السبب السابق وهو الترتيب الأساسي لا يدخل كعنصر من عناصر مجموعة التباديل . في هذا البحث لا نحتاج كامل الرقم بنظام المفكوك بل نجري عملية التحويل لحين الحصول على اصغر مرتبة تحتوي على رقم اكبر من الصفر وان أساس نظام الأرقام لهذهِ المرتبة هو X وهذهِ العملية الوحيدة التي نستخدم فيها الجملة الشرطية وتحمل Label 3 في البرنامج المرفق مع البحث . هنا يجب الإشارة إلى أن الاستبدال الرئيسي هو المرادف للجملة الشرطية الناجحة لان كل جملة شرطية ناجحة تقابلها عملية استبدال رئيسية . النقطة الأكثر أهمية بعد تحديد موقع العنصر الرئيسي الذي سيستبدل هو إلية عملية الاستبدال وهي عملية استبدال غير منفصلة non – disjoint ويمكن التعبير عنها رياضياً بالشكل التالي أذا كانت X زوجية :

$$f = \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 & X-2 \end{pmatrix} \begin{pmatrix} 1 & X-4 \end{pmatrix} \quad \ldots \quad \begin{pmatrix} 1 & 2 \end{pmatrix}$$

و التالي أذا كانت X فردية :

$$f = \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 & X-2 \end{pmatrix} \begin{pmatrix} 1 & X-4 \end{pmatrix} \quad \ldots \quad \begin{pmatrix} 1 & 3 \end{pmatrix}$$

من الواضح أن X هو رقم موقع العنصر الرئيسي الذي سيستبدل أولاً مع العنصر بالموقع رقم واحد ثم يتبع ذلك استبدالات ثانوية فإذا كان X رقم زوجي واكبر من 2 سوف نستبدل كل عنصر يقع بموقع زوجي يحمل رقم موقع أقل من X سوف يستبدل مع العنصر بالموقع الأول نزولاً من الرقم الأكبر إلى الأصغر وهو 2 إما إذا كان X رقم فردي واكبر من 3 سوف نستبدل كل عنصر يقع بموقع فردي يحمل رقم موقع أقل من X سوف يستبدل مع العنصر بالموقع الأول نزولاً من الرقم الأكبر إلى الأصغر وهو 3 وهنا نعيد التأكيد أن عملية الاستبدال نسبية أي تجرى على الترتيب السابق وليس على الترتيب الأساسي (الابتدائي) .

في الجداول السبعة التالية توضيح لعدد حالات الاستبدال الرئيسية والكلية ونقصد بالكلية هو عدد حالات الاستبدال الرئيسية والثانوية وعدد حالات استخدام الجمل الشرطية الفاشلة .

الجدول رقم (7) عندما n = 4

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
15	23	11	12	2
8	11	3	8	3
3	3	0	3	4
26	37	14	23	المجموع

الجدول رقم (8) عندما n = 5

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
75	119	59	60	2
44	59	19	40	3
15	19	4	15	4
4	4	0	4	5
138	201	82	119	المجموع

الجدول رقم (9) عندما n = 6

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
455	719	359	360	2
264	359	119	240	3
95	119	29	90	4
24	29	5	24	5
5	5	0	5	6
843	1231	512	719	المجموع

الجدول رقم (10) عندما n = 7

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
3185	5039	2519	2520	2
1854	2519	839	1680	3
665	839	209	630	4
174	209	41	168	5
35	41	6	35	6
6	6	0	6	7
5919	8653	3614	5039	المجموع

الجدول رقم (11) عندما n = 8

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
25487	40319	20159	20160	2
14832	20159	6719	13440	3
5327	6719	1679	5040	4
1392	1679	335	1344	5
287	335	55	280	6
48	55	7	48	7
7	7	0	7	8
47380	69273	28954	40319	المجموع

الجدول رقم (12) عندما n = 9

الاستبدالات الكلية	الاختبارات الكلية	الاختبارات الفاشلة	الاختبارات الناجحة (الاستبدال الرئيسي)	X
229383	362879	181439	181440	2
133496	181439	60479	120960	3
47943	60479	15119	45360	4
12536	15119	3023	12096	5
2583	3023	503	2520	6
440	503	71	432	7
63	71	8	63	8
8	8	0	8	9
426452	623521	260642	362879	المجموع

الجدول رقم (13) عندما **n = 10**

X	الاختبارات الناجحة (الاستبدال الرئيسي)	الاختبارات الفاشلة	الاختبارات الكلية	الاستبدالات الكلية
2	1814400	1814399	3628799	2293839
3	1209600	604799	1814399	1334960
4	453600	151199	604799	479439
5	120960	30239	151199	125360
6	25200	5039	30239	25839
7	4320	719	5039	4400
8	630	89	719	639
9	80	9	89	80
10	9	0	9	9
المجموع	3628799	2606492	6235291	4264565

جميع عمليات الاستبدال الرئيسية والثانوي تنفذ بالجملة الحلقية التي تتبع الجملة الشرطية المشار إليها أعلاه في البرنامج المرفق مع البحث . يمكن الحصول على قيم العمود الثاني من خلال الصيغة الرياضية التالية وبدلالة كل من n و X .

إما قيم العمود الثالث فيمكن الحصول عليها من خلال الصيغة الرياضية التالية وبدلالة كل من n و X .

$$_{n}P_{X}\ (X-1) \qquad ..... \quad (3)$$

$$_{n}P_{X}-1 \qquad ..... \quad (4)$$

إما قيم العمود الرابع وهي حاصل جمع قيم العمودين الثاني والثالث فيمكن الحصول عليها من خلال الصيغة الرياضية التالية وبدلالة كل من n و X .

$$_{n}P_{X}\ X-1 = {}_{n}P_{X-1}-1 \qquad ..... \quad (5)$$

من الملاحظ في الجداول السبعة أعلاه مجموع قيم العمود الثاني من I إلى n تساوي $_{n}P_{I-1}-1$ أي قيمة I – 1 في العمود الثالث هي .

$$\sum_{X=I}^{n} {}_{n}P_{I}\ (I-1) = {}_{n}P_{I-1}-1 \qquad ..... \quad (6)$$

ومن المعادلة 6 مجموع قيم العمود الثاني هي .

$$_{n}P_{1}-1 = n!-1 \qquad ..... \quad (7)$$

لنفرض λ تمثل مجموع قيم العمود الرابع وهو العدد الكلي لاستخدام جملة IF Statement في الجداول السبعة أعلاه حيث مجموع المعادلة (5) هو

$$\sum_{X=1}^{n-1} \frac{n!}{X!} - n + 1 = \lambda \qquad ..... \quad (8)$$

عند تبسيط المعادلة (8) نحصل على (9)

$$\sum_{X=1}^{n-1} \frac{1}{X!} = \frac{\lambda+n-1}{n!} \qquad ..... \quad (9)$$

وعند قيم $n \geq 10$ نحصل على المعادلة (10)

$$n!(e-1) \approx \lambda \qquad ..... \quad (10)$$

أشارة كل ترتيب نعتمد في حسابها على قيمة X فإذا كانت X فردية تكون $g=\frac{X-1}{2}$ وبالتالي تكون إشارة الترتيب الحالي هو حاصل ضرب إشارة الترتيب السابق في $(-1)^g$. وإذا كانت X زوجية تكون $g=\frac{X}{2}$ وبالتالي تكون إشارة الترتيب الحالي هو حاصل ضرب إشارة الترتيب السابق بـــ $(-1)^g$ علماً بان إشارة الترتيب الأساسي (الابتدائي) موجبة دائماً .

الجدولين رقم (14) و (15) المرفقة مع البحث توضح ما ذكرناه في الجانب التطبيقي حيث يوضح الجدول رقم (14) التباديل لمجموعة مكونة من 4 عناصر والأخر لــ 5 عناصر على التوالي . يتكون الجدولان من عدة أعمدة أول عمود من جهة اليسار وعنوانهُ الرقم العشري وهو مخصص لتسلسل الترتيب والتي تأخذ القيم من صفر إلى 1 - ! n وتليها مجموعة أعمدة نحول فيها الرقم في العمود الأول والذي هو بالنظام العشري إلى الرقم بصيغة المفكوك Factorial number إما العمود التالي فيحدد اصغر مرتبة لا تحتوي صفر والعمود التالي يضيف واحد إلى العمود السابق لتحديد موقع العنصر الذي ستبدأ بهِ عملية الاستبدال ويسمى X وتليه مجموعة أعمدة توضح عملية الاستبدال وتم وضع خط تحت العناصر المستبدلة علماً إن الاستبدال يحصل على الترتيب السابق وليس على الترتيب الأساسي (أو الابتدائي أو المستقل أو الافتراضي) والذي تكون فيهِ العناصر بالترتيب التالي (n , ... , 3 , 2 , 1) وكل العناصر التي تحتها خط قد تم استبداله مع العنصر في الموقع الأول ابتداً من اعلي موقع نزولاً إلى الأدنى وعلى التوالي أي كل العناصر تم استبدالها مع العنصر في الموقع الأول فقط إما العمود الأخير في الجدول فيحسب الإشارة لكل ترتيب وبالطريقة التي تم توضيحها سابقاً علماً بان الترتيب الأساسي يحمل الإشارة الموجبة دائماً .

لقد تم اختبار هذهِ الطريقة للتباديل في حساب قيمة المحدد Determinant وثبت نجاحها وكذلك سرعة التنفيذ وذلك لسببين الأول يتمثل في قلة استخدام الجمل الشرطية كما مبين سابقاً والثاني في طريقة حساب الإشارة لكل ترتيب لأعمدة المحدد فالطريقة التقليدية لحساب الإشارة تتطلب المقارنة بين أرقام الأعمدة وهذا يستغرق من الوقت بقدر الوقت المستغرق لحساب التباديل بالطريقة التقليدية وبالتالي سوف يتضاعف زمن التنفيذ بينما بطريقة البحث سوف تحتسب الإشارة من خلال معادلة لا تستغرق سوى 1% من زمن تنفيذ البرنامج المعد بطريقة البحث وقد تم مقارنة النتائج مع برامج أخرى لحساب المحدد منها الجاهز و منها مأخوذ من المصادر وثبت نجاحهُ بالإضافة إلى السرعة والدقة في التنفيذ .

5 – الاستنتاجات The deduction

بالرغم من نجاح هذهِ الطريقة وتعتبر الأسهل ألا أن هناك طرق أخرى للحصول على التباديل Permutations فبعد تحديد قيمة X يمكن إتباع إحدى الطرق التالية لإجراء عملية الاستبدال .

فإذا كانت X زوجية :

$$f=\begin{pmatrix}1 & 2\end{pmatrix}\begin{pmatrix}1 & 4\end{pmatrix}\begin{pmatrix}1 & 6\end{pmatrix}\ \ldots\ \begin{pmatrix}1 & X-4\end{pmatrix}\begin{pmatrix}1 & X-2\end{pmatrix}\begin{pmatrix}1 & X\end{pmatrix}$$

و أذا كانت X فردية :

$$f=\begin{pmatrix}1 & 3\end{pmatrix}\begin{pmatrix}1 & 5\end{pmatrix}\begin{pmatrix}1 & 7\end{pmatrix}\ \ldots\ \begin{pmatrix}1 & X-4\end{pmatrix}\begin{pmatrix}1 & X-2\end{pmatrix}\begin{pmatrix}1 & X\end{pmatrix}$$

والطريقة الأخرى للاستبدال هي فيما إذا كانت X زوجية :

$$f=\begin{pmatrix}1 & 2\end{pmatrix}\begin{pmatrix}2 & 4\end{pmatrix}\begin{pmatrix}4 & 6\end{pmatrix}\ \ldots\ \begin{pmatrix}X-4 & X-2\end{pmatrix}\begin{pmatrix}X-2 & X\end{pmatrix}=\begin{pmatrix}1 & 2 & 4 & \ldots. & X-2 & X\end{pmatrix}$$

و أذا كانت X فردية :

$$f=\begin{pmatrix}1 & 3\end{pmatrix}\begin{pmatrix}3 & 5\end{pmatrix}\begin{pmatrix}5 & 7\end{pmatrix}\ \ldots\ \begin{pmatrix}X-4 & X-2\end{pmatrix}\begin{pmatrix}X-2 & X\end{pmatrix}=\begin{pmatrix}1 & 3 & 5 & \ldots. & X-2 & X\end{pmatrix}$$

وقد تم اختبار طرق الاستبدال أعلاه وثبت نجاحها ولكن طريقة البحث هي الأسهل برمجياً وهذا يدل على المرونة العالية لهذهِ الطريقة ومع ذلك عندما تزداد قيمة n وبالتالي تزداد قيمة n! ويترتب على ذلك زيادة في

قيمة λ في المعادلة (10) والتي تمثل عدد مرات استخدام الجملة الشرطية ويترتب على ذلك زيادة زمن تنفيذ البرنامج ولهذهِ المشكلة عدة حلول فالحل المثالي هو إيجاد قيمة X من الرقم العشري من خلال معادلة بدون استخدام الجمل الشرطية ولعدم التوصل لحد ألان لمثل هذهِ المعادلة فهناك طرق أخرى منها عمل دالة تكون جزء من دوال لغة البرمجة المستخدمة بحيث تكون input هو الرقم العشري و output هو الرقم X وهذهِ الدالة أذا ما وجدت سوف تقلل من زمن تنفيذ البرنامج بشكل محدود . ومع ذلك فان لمرونة الطريقة العالية هناك عدة معالجات عندما تكون n كبيرة منها حفظ أخر ترتيب وتسلسلهُ وإشارته وقيمة المحدد Determinant داخل ملف ثم استرجاع تلك البيانات من الملف وتكملة تنفيذ البرنامج من تلك النقطة . أو حساب قيمة X في برنامج منفصل وعلى أكثر من حاسبة ثم حفظ القيم داخل ملف واستدعاء الملف في البرنامج .

جدول رقم (14) عندما n = 4

الرقم العشري	الرقم بصيغة المفكوك			المرتبة المتغيرة	الموقع المتغير	مواقع العناصر				أشارة المحدد
	1	2	3			1	2	3	4	
0	0	0	0	0	X	1	2	3	4	1
1	1	0	0	1	2	2	1	3	4	-1
2	0	1	0	2	3	3	1	2	4	1
3	1	1	0	1	2	1	3	2	4	-1
4	0	2	0	2	3	2	3	1	4	1
5	1	2	0	1	2	3	2	1	4	-1
6	0	0	1	3	4	2	4	1	3	-1
7	1	0	1	1	2	4	2	1	3	1
8	0	1	1	2	3	1	2	4	3	-1
9	1	1	1	1	2	2	1	4	3	1
10	0	2	1	2	3	4	1	2	3	-1
11	1	2	1	1	2	1	4	2	3	1
12	0	0	2	3	4	4	3	2	1	1
13	1	0	2	1	2	3	4	2	1	-1
14	0	1	2	2	3	2	4	3	1	1
15	1	1	2	1	2	4	2	3	1	-1
16	0	2	2	2	3	3	2	4	1	1
17	1	2	2	1	2	2	3	4	1	-1
18	0	0	3	3	4	3	1	4	2	-1
19	1	0	3	1	2	1	3	4	2	1
20	0	1	3	2	3	4	3	1	2	-1
21	1	1	3	1	2	3	4	1	2	1
22	0	2	3	2	3	1	4	3	2	-1
23	1	2	3	1	2	4	1	3	2	1

جدول رقم (15) عندما $n = 5$

الرقم العشري	الرقم بصيغة المفكوك				المرتبة المتغيرة	الموقع المتغير	مواقع العناصر					أشارة المحدد
	1	2	3	4			1	2	3	4	5	
0	0	0	0	0	0	X	1	2	3	4	5	1
1	1	0	0	0	1	2	2	1	3	4	5	-1
2	0	1	0	0	2	3	3	1	2	4	5	1
3	1	1	0	0	1	2	1	3	2	4	5	-1
4	0	2	0	0	2	3	2	3	1	4	5	1
5	1	2	0	0	1	2	3	2	1	4	5	-1
6	0	0	1	0	3	4	2	4	1	3	5	-1
7	1	0	1	0	1	2	4	2	1	3	5	1
8	0	1	1	0	2	3	1	2	4	3	5	-1
9	1	1	1	0	1	2	2	1	4	3	5	1
10	0	2	1	0	2	3	4	1	2	3	5	-1
11	1	2	1	0	1	2	1	4	2	3	5	1
12	0	0	2	0	3	4	4	3	2	1	5	1
13	1	0	2	0	1	2	3	4	2	1	5	-1
14	0	1	2	0	2	3	2	4	3	1	5	1
15	1	1	2	0	1	2	4	2	3	1	5	-1
16	0	2	2	0	2	3	3	2	4	1	5	1
17	1	2	2	0	1	2	2	3	4	1	5	-1
18	0	0	3	0	3	4	3	1	4	2	5	-1
19	1	0	3	0	1	2	1	3	4	2	5	1
20	0	1	3	0	2	3	4	3	1	2	5	-1
21	1	1	3	0	1	2	3	4	1	2	5	1
22	0	2	3	0	2	3	1	4	3	2	5	-1
23	1	2	3	0	1	2	4	1	3	2	5	1

الرقم العشري	الرقم بصيغة المفكوك				المرتبة المتغيرة	الموقع المتغير	مواقع العناصر					أشارة المحدد
	1	2	3	4			1	2	3	4	5	
24	0	0	0	1	4	5	3	1	5	2	4	1
25	1	0	0	1	1	2	1	3	5	2	4	-1
26	0	1	0	1	2	3	5	3	1	2	4	1
27	1	1	0	1	1	2	3	5	1	2	4	-1
28	0	2	0	1	2	3	1	5	3	2	4	1
29	1	2	0	1	1	2	5	1	3	2	4	-1
30	0	0	1	1	3	4	1	2	3	5	4	-1
31	1	0	1	1	1	2	2	1	3	5	4	1
32	0	1	1	1	2	3	3	1	2	5	4	-1
33	1	1	1	1	1	2	1	3	2	5	4	1
34	0	2	1	1	2	3	2	3	1	5	4	-1
35	1	2	1	1	1	2	3	2	1	5	4	1
36	0	0	2	1	3	4	2	5	1	3	4	1
37	1	0	2	1	1	2	5	2	1	3	4	-1
38	0	1	2	1	2	3	1	2	5	3	4	1
39	1	1	2	1	1	2	2	1	5	3	4	-1
40	0	2	2	1	2	3	5	1	2	3	4	1
41	1	2	2	1	1	2	1	5	2	3	4	-1
42	0	0	3	1	3	4	5	3	2	1	4	-1
43	1	0	3	1	1	2	3	5	2	1	4	1
44	0	1	3	1	2	3	2	5	3	1	4	-1
45	1	1	3	1	1	2	5	2	3	1	4	1
46	0	2	3	1	2	3	3	2	5	1	4	-1
47	1	2	3	1	1	2	2	3	5	1	4	1
48	0	0	0	2	4	5	5	3	4	1	2	1

الرقم العشري	الرقم بصيغة المفكوك				المرتبة المتغيرة	الموقع المتغير	مواقع العناصر					أشارة المحدد
	1	2	3	4			1	2	3	4	5	
49	1	0	0	2	1	2	3	5	4	1	2	-1
50	0	1	0	2	2	3	4	5	3	1	2	1
51	1	1	0	2	1	2	5	4	3	1	2	-1
52	0	2	0	2	2	3	3	4	5	1	2	1
53	1	2	0	2	1	2	4	3	5	1	2	-1
54	0	0	1	2	3	4	3	1	5	4	2	-1
55	1	0	1	2	1	2	1	3	5	4	2	1
56	0	1	1	2	2	3	5	3	1	4	2	-1
57	1	1	1	2	1	2	3	5	1	4	2	1
58	0	2	1	2	2	3	1	5	3	4	2	-1
59	1	2	1	2	1	2	5	1	3	4	2	1
60	0	0	2	2	3	4	1	4	3	5	2	1
61	1	0	2	2	1	2	4	1	3	5	2	-1
62	0	1	2	2	2	3	3	1	4	5	2	1
63	1	1	2	2	1	2	1	3	4	5	2	-1
64	0	2	2	2	2	3	4	3	1	5	2	1
65	1	2	2	2	1	2	3	4	1	5	2	-1
66	0	0	3	2	3	4	4	5	1	3	2	-1
67	1	0	3	2	1	2	5	4	1	3	2	1
68	0	1	3	2	2	3	1	4	5	3	2	-1
69	1	1	3	2	1	2	4	1	5	3	2	1
70	0	2	3	2	2	3	5	1	4	3	2	-1
71	1	2	3	2	1	2	1	5	4	3	2	1
72	0	0	0	3	4	5	4	5	2	3	1	1
73	1	0	0	3	1	2	5	4	2	3	1	-1

الرقم العشري	الرقم بصيغة المفكوك				المرتبة المتغيرة	الموقع المتغير	مواقع العناصر					أشارة المحدد
	1	2	3	4			1	2	3	4	5	
74	0	1	0	3	2	3	2	4	5	3	1	1
75	1	1	0	3	1	2	4	2	5	3	1	-1
76	0	2	0	3	2	3	5	2	4	3	1	1
77	1	2	0	3	1	2	2	5	4	3	1	-1
78	0	0	1	3	3	4	5	3	4	2	1	-1
79	1	0	1	3	1	2	3	5	4	2	1	1
80	0	1	1	3	2	3	4	5	3	2	1	-1
81	1	1	1	3	1	2	5	4	3	2	1	1
82	0	2	1	3	2	3	3	4	5	2	1	-1
83	1	2	1	3	1	2	4	3	5	2	1	1
84	0	0	2	3	3	4	3	2	5	4	1	1
85	1	0	2	3	1	2	2	3	5	4	1	-1
86	0	1	2	3	2	3	5	3	2	4	1	1
87	1	1	2	3	1	2	3	5	2	4	1	-1
88	0	2	2	3	2	3	2	5	3	4	1	1
89	1	2	2	3	1	2	5	2	3	4	1	-1
90	0	0	3	3	3	4	2	4	3	5	1	-1
91	1	0	3	3	1	2	4	2	3	5	1	1
92	0	1	3	3	2	3	3	2	4	5	1	-1
93	1	1	3	3	1	2	2	3	4	5	1	1
94	0	2	3	3	2	3	4	3	2	5	1	-1
95	1	2	3	3	1	2	3	4	2	5	1	1
96	0	0	0	4	4	5	2	4	1	5	3	1
97	1	0	0	4	1	2	4	2	1	5	3	-1
98	0	1	0	4	2	3	1	2	4	5	3	1

الرقم العشري	الرقم بصيغة المفكوك				المرتبة المتغيرة	الموقع المتغير	مواقع العناصر					أشارة المحدد
	1	2	3	4			1	2	3	4	5	
99	1	1	0	4	1	2	2	1	4	5	3	-1
100	0	2	0	4	2	3	4	1	2	5	3	1
101	1	2	0	4	1	2	1	4	2	5	3	-1
102	0	0	1	4	3	4	4	5	2	1	3	-1
103	1	0	1	4	1	2	5	4	2	1	3	1
104	0	1	1	4	2	3	2	4	5	1	3	-1
105	1	1	1	4	1	2	4	2	5	1	3	1
106	0	2	1	4	2	3	5	2	4	1	3	-1
107	1	2	1	4	1	2	2	5	4	1	3	1
108	0	0	2	4	3	4	5	1	4	2	3	1
109	1	0	2	4	1	2	1	5	4	2	3	-1
110	0	1	2	4	2	3	4	5	1	2	3	1
111	1	1	2	4	1	2	5	4	1	2	3	-1
112	0	2	2	4	2	3	1	4	5	2	3	1
113	1	2	2	4	1	2	4	1	5	2	3	-1
114	0	0	3	4	3	4	1	2	5	4	3	-1
115	1	0	3	4	1	2	2	1	5	4	3	1
116	0	1	3	4	2	3	5	1	2	4	3	-1
117	1	1	3	4	1	2	1	5	2	4	3	1
118	0	2	3	4	2	3	2	5	1	4	3	-1
119	1	2	3	4	1	2	5	2	1	4	3	1

Reference المصادر

1 - Biswas,S. (1984) "Topical In Algebra Of Matrices " . India . Academic Publication .

2 - Feller,W. (1988) "An Introduction To Probability Theory And It's Application" .Third Edition .New York. John Wiley & Sons .

3 - Gilbert , William J & W. Keith Nicholson . (2004) " Modern Algebra With Applications " . Second Edition . New Jersey . John Wiley & Sons .

4 - Gathen , Joachim Von Zur & JÜrgen Gerhard . (1999) "Modern Computer Algebra" . United Kingdom . Cambridge University Press .

5 - Kintchine,A.Y. (1960) "Mathematical Method In The Theory Of Queuing" .London. Chrler Grissin And Company Limited .

6 - Lining,J.H. & R.H.Battin. (1956) "Random Processes" . New York. McGraw-Hill.

7 - Parzen,E. (1960) "Modern Probability Theory And It's Application". New York. John Wiley & Sons .

8 - Phillips,D.T. (1976) "Operation Research Principals And Practices". New York. John Wiley & Sons .

9 - Snall,G.L. (1989) "Introduction To Probability". New York. McGraw-Hill.

10 - Taha,H.A. (1976) " Operation Research An Introduction". Second Edition. London. Collier Macmillan.


```
' this program calculate determinant by Rafid Fayadh method

'read data of matrix

CLS
INPUT numb
DIM af(numb, numb) AS DOUBLE

        FOR i = 1 TO numb
                FOR j = 1 TO numb
                        READ af(i, j)
                NEXT
        NEXT


        'find factorial

g1 = 1: FOR e = 1 TO numb: g1 = g1 * e: NEXT
PRINT g1

        ' find first item of determinant

DIM h AS DOUBLE
DIM bf(numb) AS INTEGER: h = 0: m = 0
FOR e = 1 TO numb: bf(e) = e: NEXT
```

```
'
            g = 1: FOR e = 1 TO numb
            g = g * af(e, bf(e)): NEXT: h = h + g * (-1) ^ m

      ' find remain  items of determinant

 FOR i = 1 TO g1 - 1
      f = 2: s = i
      ' line label 3 to convert decimal number to factorial number
3 IF s / f = INT(s / f) THEN s = s / f: f = f + 1: GOTO 3

 ' after determine f this loop is the core of program

      FOR p = f TO 2 STEP -2
      m = m + 1
      sw = bf(1): bf(1) = bf(p): bf(p) = sw
      NEXT

            g = 1: FOR e = 1 TO numb
            g = g * af(e, bf(e)): NEXT: h = h + g * (-1) ^ m

      FOR y=1 to numb : PRINT bf(y);:NEXT: PRINT
 NEXT

 ' print value of determinant

 PRINT h

DATA 43,  56,  12,  33 , 5, 56
DATA 98,  61,  77,  16 , 8, 62
DATA 52 , 78,  52,  48 , 9, 81
DATA 62 , 39,  38 , 31 , 2, 14
DATA 6, 4, 6, 9, 4, 33
DATA 7, 11,  9, 5, 3, 64
```

....

....

....