تصميم وتنفيذ
الجزء التحليلي لمؤلف
لغة برمجة عربية
على نمط لغة باسكال

الدكتور حسن لعيبي ناصر
الجامعة التكنولوجية / قسم علوم الحاسبات

محمود قاسم شريف

* من البحوث المقدمة في مؤتمر بغداد العالمي الثاني لتقنيات وتطبيقات الحاسبات الالكترونية

الخلاصة :

يهدف البحث الى تصميم وتنفيذ لغة برمجة عربية تعليمية وقع الاختيار على لغة البرمجة العليا باسكال لكي تكون اللغة العربية المصممة شبيهة بها , وذلك لان هذه اللغة توفر البرمجة الهيكلية STRUCTURED PROGRAMMING اضافة لكونها لغة تعليمية . وعليه فقد وضعت قواعد اللغة موضوعة البحث , كمجموعة جزئية من قواعد لغة باسكال PASCAL SUBSET وعرفت بصيغة BACKUS-NAUR FORM BNF بعد تعريف قواعد اللغة , اتجه البحث لبناء مراحل البرنامج المؤلف COMPILER والذي يقوم بترجمة اللغة المقترحة الى لغة الماكنة MACHINE LANGUAGE .

وقد تم بناء المراحل التالية :-

أولا- محلل المفردات او الماسح LEXICAL ANALYSER OR SCANNER

ثانيا- محلل القواعد او المعرب SYNTAX ANALYSER OR PARSER

ثالثا- محلل المعنى SEMANTIC ANALYSER

استخدمت في بناء المحلل النحوي طريقة الانحدار التكراري الذاتي للاعراب RECURSIVE DECENT COMPILING .

عرفت مجموعة من رسائل الاخطاء في كل من محلل القواعد ومحلل المعنى يقوم كلا المحللان بتحليل برامج المستفيد بعد قراءتها وتدقيق قواعدها مع قواعد اللغة المصممة , فأن كانت خالية

من الاخطاء ، عندئذ تكتب على ملف خارجي تمهيدا لاستدعاء البرنامج المؤلف للغة باسكال COMPILIER PASCAL الذي يقوم بتنفيذها . وبخلافه ، اي عند وجود اخطاء يقوم بأعلان جمل الاخطاء للمستفيد لكي يقوم بتصليحها .

تم فحص المراحل المنفذة من البرنامج المؤلف على مجموعة من برامج المستفيدين وكانت النتائج جيدة ، حيث كتبت البرامج باللغة العربية ومن وحدة الادخال القياسية STANDARD INPUT ونفذت من قبل الحاسبة NEC-350 المتوفرة في معهد التدريب والبحوث وهي الحاسبة التي اجري عليها البحث والذي كتب بلغة البرمجة العليا باسكال . وكما نلاحظ من النتيجة المرفقة بالتقرير ، فأن البرنامج المؤلف يحتوي على خاصية الرجوع من الخطأ .

المقدمة :

كانت اللغة العربية لغة الثقافة والعلم في اوج عظمة الحضارة الاسلامية ولاتزال اليوم تسير بخطى حثيثه الى الامام لمواكبة العصر والتطور التكنولوجي معززة اصالتها وقدرتها ووجودها الحي والفاعل بين الامم الناهضة والمتطورة .

وحيث ان الحاسبات الالكترونية تعتبر انجازا متقدما للمسيرة التكنولوجية والعلمية بفعل الجهد البشري بحيث لم تبق زاوية من زوايا الحاجات الانسانية المتعددة الا ودخلتها الحاسبات الالكترونية من اوسع ابوابها .

وحيث ان المستفيد يتعامل مع الحاسبات الالكترونية عبر اللغة التي تفهمها الحاسبة الالكترونية ، كانت الحاجة الملحة تستدعي ايجاد لغة عربية تتعامل مع هذه التقنية يسهل تداولها من قبل المستفيد العربي .

ان هذا البحث يقع ضمن فعاليات التعريب في هذا الميدان .

توخينا في هذا البحث تجنب السلبيات الواردة في البحوث السابقة ، حيث يمكن للمستفيد ان يكتب البرامج بصيغة حرة FREE FORMAT كما يمكن للمستفيد ان يكتب برامج مساعدة معرفة داخليا NESTED PROCEDURES وحسب ما تسمح به الحاسبة التي يطبق عليها البرنامج .

يمكن ملاحظة ميزات اللغة من خلال مراجعة تعريف قواعدها في الملحق .

نبذة عامة عن البرنامج المؤلف COMPILER

ان البرنامج المؤلف عبارة عن برنامج مكتوب بلغة برمجة معينة ، يقوم بترجمة برنامج المستفيد الذي يعرف بالبرنامج المصدري SOURCE PROG الى برنامج الهدف OBJECT PROGRAM .

يكتب البرنامج المصدري بلغة يفهمها المستفيد بسهولة وتسمى لغة المصدر SOURCE LANGUAGE ، بينما يكتب برنامج الهدف بلغة تفهمها الحاسبة تسمى لغة الماكنة MACHINE LANGUAGE ، وهذه اللغة نوعا ما صعبة على المستفيد .

باختصار ، يمكن تعريف البرنامج المؤلف بأنه اداة تقوم بتحويل من ميدان المستفيد لحل المشاكل الى ميدان الماكنة لتنفيذ الحلول وبدون اي تغيير للمعنى او المضمون لهذه البرامج .

مراحل البرنامج المؤلف COMPILER PHASES

من الاساليب المعتمدة في تصميم نظم البرامجيات هي تجزئة المشكلة موضوعة البحث الى مشاكل ثانوية ، يمكن معالجة كل واحدة من هذه المشاكل باستخدام برنامج اولي بسيط MODULE . في الشرح الموجز التالي ، سنلزم انفسنا باعلى مستويات التجزيء لعملية التأليف TOP LEVEL OF REFINEMENT. وهذه

المستويات والتي تسمى مراحل تسمى PHASES هي عامة لاغلب البرامج المؤلفة .

يقوم البرنامج المؤلف بتحليل البرنامج المصدري وتركيب برنامج الهدف عمليا فأن البرامج المؤلفة تقوم بعملية التحليل بمرحلتين منفصلتين :

تحليل البرنامج الى وحدات اساسية LEXICAL ANALYSIS :

يتكون برنامج المستفيد من سلسلة من الاحرف تقرأ بواسطة المرحلة الاولى من مراحل البرنامج المؤلف ، اي مرحلة تحليل البرنامج الى وحدات اساسية . تقوم هذه المرحلة بتحويل هذا المظهر الخارجي للبرنامج المصدري الى تمثيل داخلي INTERNAL FORMAT اكثر ملاءمة للمعالجة من قبل المراحل الاخرى للبرنامج المؤلف .

تتلخص المهام الرئيسية لمحلل المفردات فيما يلي :-

(1) التعرف على الوحدات الرئيسية للبرنامج BASIC LEXICAL UNIT والتي تسمى TOKENS وذلك بدون الرجوع الى اي علاقات بين هذه الوحدات او فيما يختص بمعانيها .

(2) ازالة كافة الفراغات في البرنامج المصدري وكذلك كافة الخواص التي تميز وسط الدخل INPUT MEDIUM .

(3) ازالة الملاحظات من البرنامج المصدري .

(4) اعطاء بعض جمل الاخطاء عند اكتشافها .

تحليل القواعد SYNTAX ANALYSIS :

التحليل القواعدي هو المرحلة الاخرى من مراحل التحليل . تتميز كل لغة برمجة عن لغات البرمجة الاخرى بمجموعة من قواعد اللغة خاصة بها وتميزها عن لغات البرمجة الاخرى بحيث تميز هذه القواعد البرامج الصحيحة التي تكتب بتلك اللغة ، وبالمقارنة مع اللغة الطبيعية NATURAL LANGUAGE ، فأن هذه القواعد تشبه الفعاليات في ايجاد كل من الفعل والفاعل والاسم والصفة في جمل معينة عند اعرابها . يقوم محلل القواعد باستلام خرج محلل المفردات كدخل له بعد ان يتحقق من ان قواعد البرنامج المصدري متطابقة مع الشروط القواعدية التي تسمح بها لغة البرمجة ، موضوع البحث ، اما الخرج لهذه المرحلة فهو ما يسمى بشجرة الاعراب PARSE TREE وهي عبارة عن تمثيل داخلي للبرنامج وبهيكل معين يسمح للمراحل التالية من البرنامج المؤلف بالتعرف على العلاقات بين اجزاء البرنامج بعضها البعض وارتباطها بالبرنامج ككل .

تحليل المعنى او المضمون SEMANTIC ANALYSIS :

تحتوي اغلب لغات البرمجة على ميزات لايمكن توضيحها عند تعريفها بصيغة BNF بسبب الحدود العرفية التي تكتنف هذه الصيغة . يقوم محلل المعنى بتقصي هذه الميزات . كل نواحي اللغة المعرفة في القواعد تسمى قواعدية وكل النواحي الاخرى للغة هي مضمونة SEMANTIC فمثلا لايمكن التوضيح في قواعد اللغة بأن متغير ما يجب ان يعرف مرة وحدة فقط اذا تم الرجوع اليه او استعمل في جملة احلال ، ولكن محلل المعنى يمكنه التحقق من انتهاك هذه القاعدة .

توليد الجفرة CODE GENERATION :

تعتبر هذه المرحلة بناء اكثر منها تحليل ، وهي تأخذ شجرة الاعراب PARSE TREE وتبدأ بتعقبها TRAVERSE واعتمادا على العلاقات الهيكلية التي تميزها تبدأ بتوليد شفرة الهدف OBJECT CODE . نتوقف عند هذه المرحلة من مراحل البرنامج المؤلف ، وذلك لان المراحل التي ذكرت حتى الان تعتبر متشابهة بين اغلب البرامج المؤلفة . بعد اعطاء هذه الفكرة الموجزة عن مراحل البرنامج المؤلف ، سنقوم فيما يلي بسرد شرح موجز للانجاز العملي ، وسنركز على الطريقة التي اعتمدت في بناء

محلل القواعد والتي يطلق عليها اسم : الانحدار التكـــــراري الذاتي وذلك لانها اهم مرحلة من مراحل البرنامج المؤلف .

التأليف بالانحدار التكراري الذاتي

RECURSIVE DECENT COMPILING

سيرد فيما يلي شرح لتقنية محددة ، وهي المذكورة اعـــــلاه ، وتطبيقها على لغة البرمجة العربية التعليمية المقترحة ،وهي لغة تمثل مجموعة جزئية من لغة البرمجة العليا PASCAL .

ما المقصود بالانحدار التكراري الذاتي ؟

تركز هذه التقنية على مرحلة تحليل القواعد للبرنامـــــــج المؤلف ، بحيث تقسم مرحلة تحليل القواعد الى مجموعـــــة من البرامـــج المساعـــدة التي تقـــوم بعمليـــة التعـــــــرف RECOGNITION ROUTINES ، بحيث ان كل واحدة من هذه البرامـــج المساعدة عليها واجب التأكد من امكان وجود عبارة محددة في الدخل معرفة اصلا في قواعد اللغة .

وكل واحدة من هذه البرامج المساعدة يمكنها ان تستدعي برامج مساعدة اخرى ، بعد تعرفها على عبارة معينة في الدخـــــل ، وتقوم هذه البرامج الاخرى بالتعرف على عبارات اخـــــــرى SUBPHRASES تتضمنها العبارة الاصلية وهكذا حتى نهايـــــــة العبارات في الدخل .

اغلب هذه البرامج المساعدة تكون متبادلة بالتكرار MUTUALLY RECURSIVE وذلك يعكس حقيقة انه في اي نص يحتمل وجود نصوص اخرى مغمورة EMBEDDED به ، فمثلا التعريف قد يحوي تعريفات داخلية INNER DECLARATIONS وكذلك التعبير قد يحوي على تعبير داخلي SUBEXPRESSION وهكذا ...

بعض البرامج المساعدة المشخصة RECOGNIZERS يجب عليها اتخاذ قرار ما ، وفي مثل هذه الحالات ، يتم اتخاذ القرار استنادا على معرفة الرمز التالي في السطر الداخل .

ان المطلوب من البرنامج المؤلف ، ليس فقط التعرف على البرامج الصحيحة قواعديا ، بل التعرف على الجمل التي لاتتطابق مع قواعد اللغة اضافة للتحقق من التعريفات وانواعها في اللغة TYPE CHECKING وهذا ما اخذ بالاعتبار عند برمجة محلل القواعد .

اهم خصائص تقنية الاعراب هذه عند تطبيقها عمليا ، هي القيام بعملية التشخيص وتدقيق النوع وتوليد الشفرة بدون الحاجة الى ذاكرة مساعدة " BACKUP " ، بمعنى ، اذا استدعى أحد البرامج المساعدة وليكن ' أ ' برنامجا مساعدا آخر وليكن ' ب ' فأنه عندئذ يكون متأكدا ، ان كان مبرمجا بشكل صحيح ، بأنه قام بألاختيار المناسب والصحيح وذلك اعتمادا= على الدخل الذي وصله .

وعليها بفرض القيد التالي على اللغة التي يمكن ترجمتها بهذه التقنية LL (1) CONDITIONS ، حيث يعني حرف 'L' الاول عملية مسح من اليسار لليمين LEFT TO RIGHT SCANNING والحرف 'L' الثاني استخدام الاشتقاق من جهة اليسار LEFT MOST DERIVATION أما الرقم '1' فيعني أن عدد الرموز النهائية التي يمكن التكهن بها مسبقا هي واحد NUMBER OF TERMINALS OF LOOKAHEAD ان القواعد التي يتم التعرف عليها من قبل محلل القواعد نوع LL(K) تسمى LL(K)-GRAMMER ، وان اللغة التي لها وصف قواعدي من نوع LL (K) - GRAMMER تسمى LL(K)-LANGUAGE .

من الخصائص المهمة للقواعد نوع LL(1) انها لاتحوي على غموض

الاستنتاجات والاقتراحات :

يتضمن هذا الجزء ما تم انجازه في البحث مع طرح بعض التعديلات التي لم يتسع الوقت لادخالها في البحث والتي تحسن من اداءها .

قبل البداية في مراحل البحث ، اتجهنا نحو اختيار لغة برمجة عليها نكتب بها البرنامج المؤلف بأكفء شكل ممكن . وبعد فترة من الدراسة وقع الاختيار على لغة البرمجة العليا PASCAL

وذلك لما تحويه من ميزات تفصلها عن باقي اللغات الشائعة مثل BASIC , FORTRAN .

ويأتي هذا التميز لما تحويه لغة باسكال من هياكل معلوماتية DATA STRUCTURES وجمل هيكلية STRUCTURES ومجموعة من الجمل الشرطية ووجود خاصية تؤثر على كفاءة البرنامج المصمم وهي أن لغة الباسكال من نوع اللغات الموجهة نحو الحل وهذه الخاصية تعطي كفاءة عالية .

(1) يمكن بواسطة اللغة المصممة تمثيل المتغيرات بنوعين ، متغير بسيط أو على شكل مصفوفة احادية البعد ONE DIMENTIONAL ARRAY .

(2) يمكن تعريف المتغيرات باللغة بأي عدد من الاحرف وقد تتخللها الارقام ، ويقوم محلل القواعد بتمييز أول ستة احرف من المتغير أما باقي الاحرف فتميز من قبل المستفيد .

(3) تميزت اللغة بالسهولة والبساطة لكي تتلائم مع الاهداف التي صممت من اجلها ، ولذلك لم تعرف الانواع الاخرى من الهياكل والتي تحويها لغة باسكال ، ويمكن ادخال هذه التعاريف في بحوث لاحقة .

(4) يمكن كتابة كافة ايعازات البرنامج بصيغة حرة ويمكن ترك الفراغات حسب الرغبة ، بين الرموز الاساسية في اللغة وليس خلالها ويمكن كتابة عدة ايعازات على نفس السطر .

(5) لاتقيد اللغة المستفيد بترقيم السطور ، بل تكتب الايعازات بلا ترقيم ، ويقوم البرنامج بترقيمها عند الاخراج ، ليسهل على المستفيد تحديد مواضع الخطأ .

(6) يمكن تعريف عدة برامج مساعدة في البرنامج الرئيسي ويمكن تعريف متغيرات محلية لكل برنامج مساعد وحسب الحاجة .

(7) وضعت تعاريف في اللغة تختص بالعمليات المنطقية والتي امكن الاستفادة منها ببناء محلل المعنى والذي ينفذ تدقيقه على التوازي مع محلل القواعد .

يرفق بالتقرير احد البرامج التي فحص مدى تطابق قواعدها مع قواعد اللغة المصممة .

يتضح من المثال المرفق ان اللغة المصممة تسمح بظهور اربعة انواع من المتغيرات وهي متغيرات من نوع عدد صحيح ومن نوع احرف ومن نوع هامشي ومن نوع منطقي .

وعليه يمكن تدقيق اي عملية حسابية أو عملية احلال أو عملية مقارنة واعطاء جمل أخطاء ان كانت المتغيرات غير متطابقة

بالنوع .وكذلك امكن تمييز المتغير المعرف الذي يعني برنامج مساعد عن المتغير المعرف الذي يعني بداية جملة احلال وذلك عند ورودهما في البرنامج الرئيسي .

شكر وتقدير :

أجري هذا البحث على حاسبة معهد التدريب والبحوث للحاسبات والاجهزة الالكترونية ، وبهذه المناسبة يود الباحثان التعبير عن شكرهما وتقديرهما الى ادارتي المركز القومي للحاسبات ومعهد التدريب والبحوث .

المصـــــــــادر

(1) BARRON D.W
PASCAL THE LANGUAGE AND ITS IMPLEMENTATION
JOHN WIELY & SONS 1979

(2) BROWN P.J.
WRITING INTERACTIVE COMPILERS AND INTERPRETERS
JOHN WIELY & SONS 1981

(3) DAVIE A. J. T. & MORRIOSN R.
RECURSIVE DECENT COMPILING
ELLIS HORWARD LTD. PUBLICATION 1981

(4) HUNTER R.
THE DESIGN AND COSTRUCTION OF COMPILERS
JOHN WIELY & SONS 1981

(5) PYSTER A.B.
COMPILER DESIGN AND CONSTRUSTION
VAN NORTRAND REINHOLD COMP. 1980

(6) WICHMAN B. A. & CIECHANOWICZ Z. J.
PASCAL COMPILER VALIDATION
JOHN WIELY & SONS 1983

(7) WIRTH N. & JENSEN K.
PASCAL USER MANUAL AND REPORT
SPRINGER-VELAG N. Y. 1976

(8) ULLMAN J. D. & AHO A. V.
PRINCIPLES OF COMPILER DESIGN
ADDISON-WESLEY PUBLICATIONS 1977

الملاحق

ملحق (1) يبين تعريف اللغة العربية المصممة بصيغة BNF

<برنامج> ::= برنامج <معرف>،<كتلة>

<كتلة> ::= <جزء تعريف المتغيرات>|<جزء تعريف البرامج المساعدة> | <جزء العبارات>،

<جزء تعريف المتغيرات> ::= <فراغ> | متغير <تعريف المتغير>، {<تعريف المتغير>،}

<تعريف المتغير> ::= <معرف>{ ، <معرف>} : <سمة>

<سمة> ::= <سمة بسيطة> | <سمة مصفوفة>

<سمة مصفوفة> ::= مصفوفة [<مدى الفهرس>] نوع <سمة بسيطة>

<مدى الفهرس> ::= <عدد صحيح ثابت> .. <عدد صحيح ثابت>

<سمة بسيطة> ::= <سمة معرف>

<سمة معرف> ::= <معرف>

<جزء تعريف البرامج المساعدة> ::= {<تعريف البرنامج المساعد>،}

<تعريف البرنامج المساعد> ::= برنمج <معرف>،<كتلة>

<جزء العبارات> ::= <عبارة مركبة>

<جزء العبارات> ::= <عبارة مركبة>

<عبارة مركبة> ::= ابدا <عبارة>{<عبارة>} انهي

<عبارة> ::=<عبارة بسيطة> ا<عبارة مهيكلة>

<عبارة بسيطة> ::=<عبارة احلال> ا<عبارة برنامج مساعد>ا

<عبارة قراءة> ا<عبارة كتابة>

<عبارة احلال>::= <متغير> :=<تعبير>

<عبارة برنامج مساعد>::= <معرّف برنامج مساعد>

<معرّف برنامج مساعد> ::= <معرّف>

<عبارة قراءة> ::= اقرا (<متغير دخل>{ ,<متغير دخل> })

<متغير دخل>::= <متغير>

<عبارة كتابة> ::= اكتب (<قيمة خرج>{ ,<قيمة خرج> })

<قيمة خرج>::= <تعبير>

<عبارة مهيكلة>::= <عبارة مركبة> ا<عبارة اذا>ا<عبارة بينما>

<عبارة اذا>::= اذا <تعبير> عندئذ <عبارة> ا

اذا <تعبير> عندئذ <عبارة> بخلافة <عبارة>

<عبارة بينما> ::= بينما <تعبير> نفذ<عبارة>

<تعبير> ::= <تعبير بسيط>ا

<تعبير بسيط><عامل مقارنة><تعبير بسيط>

<تعبير بسيط> ::=<اشارة><حد>{<عامل جمع><حد>}

<حد> ::=<وسيط>{<عامل ضرب><وسيط>}

<وسيط> ::= <متغير> ا<ثابت>ا (<تعبير>) ا نفي<وسيط>

<عامل مقارنة> ::= = ا< >ا< ا> ا<= ا>=

<اشارة> ::= +|-|<فراغ>

<عامل جمع> ::= +|-| او

<عامل ضرب> ::= * | تقسيم | و

<متغير> ::= <متغير تام> | <متغير مفهرس>

<متغير مفهرس> ::= <متغير هامشي> [<تعبير>]

<متغير هامشي> ::= <متغير تام>

<متغير تام> ::= <متغير معرّف>

<متغير معرّف> ::= <معرّف>

<ثابت> ::= <عدد صحيح ثابت> | <حرف ثابت> | <معرّف ثابت>

<معرّف ثابت> ::= <معرّف>

<معرّف> ::= <حرف> {<حرف> | <رقم>}

<حرف> ::= ا ب ا ت ا ث ا ج ا ح ا خ ا د ا ذ ا ر ا ز

س ا ش ا ص ا ض ا ط ا ظ ا ع ا غ ا ف ا ق ا ك

ل ا م ا ن ا ه ا و ا ي ا ئ ا ة

<رقم> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

الملحق (2) يوضح احد البرامج المكتوبة باللغة المصممـــــــة وكيفية اكتشاف الاخطاء بها .

التسلسل الناتج بعد تنفيذ عملية تحليل القواعد والمضمون

```
0       برنامج فحص للمترجم :
1       ( * برنامج بسيط لفحص قواعد اللغة المصممة * )
2       متغيـــر
3       ( * فيما يلي الحقل الخاص بالتعريفات * )
4       سمة متغير : حقيقي :
*****           ^خطـــأ 144
5       سمة منطق : بولن :
*****           ^خطـــأ 144
6       سمة حرف حرف :
*****           ^خطـــــأ 123
7       سمة مصفوفة : مصفوفة 1 ..]نوع حرف ؛
*****           ^خطـــأ 119
*****           ^خطـــأ 101
8       ( * فيما يلي كتلة البرنامج * )
9       ابدا
```

```
10      اقرا (سمة متغير,سمة منطق,سمة حرف,سمة مصفوفة) ؛
*****                                          ^خطأ 152
11      اقرا ( سمة مصفوفة [5] ) ؛
*****                                          ^خطأ 117
12      اذا سمة متغير< >سمة حرف عندئذ سمة متغير:= سمة متغير+10؛
13      بينما سمة متغير < > 50 نفذ
14      ابدا
15      اكتب سمة متغير ,(سمة متغير+10) , سمة حرف ) ؛
*****                                          ^خطأ 116
16      سمة مصفوفة [ ي ] :=  سمة مصفوفة [ ي+1]+ 5:
*****                                          ^خطأ 144
*****                                          ^خطأ 149
*****                                          ^خطأ 151
17      اكتب (سمة متغير *5 + ي)
18      انهي :
19      سمة منطق :=  صح: سمة منطق :=  خطأ ؛
20      اذا سمة منطق :=  صح عندئذ سمة منطق :=  خطأ
*****                                          ^خطأ 140
*****                                          ^خطأ 106
```

21 بخلافة سمة منطق :≡ صح !

22 اكتب سمة متغير , سمة حرف

***** ^خطـــأ 116

23 انتهى

***** ^خطـــأ 117

انتهت عمليـــة تحليل القواعد والمضمـــون

عدد الاخطـــــاء المعلنة : 15

ملحـــق (3)

قائمة بمعاني ارقام الاخطاء التي يعلنها محلل القواعد :

الخطـــأ رقم 100 يعني ان الرمز المتوقع ظهوره هو معرّف

الخطـــأ رقم 101 يعني ان الرمز المتوقع ظهوره هو عدد ثابت صحيح

الخطـــأ رقم 102 يعني ان الرمز المتوقع ظهوره هو حرف ثابت

الخطـــأ رقم 103 يعني ان الرمز المتوقع ظهوره هو " نفي "

الخطـــأ رقم 104 يعني ان الرمز المتوقع ظهوره هو " و "

الخطـــأ رقم 105 يعني ان الرمز المتوقع ظهوره هو " أو "

الخطـــأ رقم 106 يعني ان الرمز المتوقع ظهوره هو " * "

الخطـــأ رقم 107 يعني ان الرمز المتوقع ظهوره هو "تقسيم"

الخطـــأ رقم 108 يعني ان الرمز المتوقع ظهوره هو " + "

الخطـــأ رقم 109 يعني ان الرمز المتوقع ظهوره هو " - "

الخطـــأ رقم 110 يعني ان الرمز المتوقع ظهوره هو " > "

الخطـــأ رقم 111 يعني ان الرمز المتوقع ظهوره هو " => "

الخطـــأ رقم 112 يعني ان الرمز المتوقع ظهوره هو " < "

الخطـــأ رقم 113 يعني ان الرمز المتوقع ظهوره هو " =< "

الخطـــأ رقم 114 يعني ان الرمز المتوقع ظهوره هو " <> "

الخطـــأ رقم 115 يعني ان الرمز المتوقع ظهوره هو " = "

الخطـــأ رقم 116 يعني ان الرمز المتوقع ظهوره هو ") "

الخطـــأ رقم 117 يعني ان الرمز المتوقع ظهوره هو ") "

الخطـــأ رقم 118 يعني ان الرمز المتوقع ظهوره هو "] "

الخطـــأ رقم 119 يعني ان الرمز المتوقع ظهوره هو " ["

الخطـــأ رقم 120 يعني ان الرمز المتوقع ظهوره هو " ، "

الخطـــأ رقم 121 يعني ان الرمز المتوقع ظهوره هو " ، "

الخطـــأ رقم 122 يعني ان الرمز المتوقع ظهوره هو " " "

الخطـــأ رقم 123 يعني ان الرمز المتوقع ظهوره هو " # "

الخطـــأ رقم 124 يعني ان الرمز المتوقع ظهوره هو " =# "

الخطـــأ رقم 125 يعني ان الرمز المتوقع ظهوره هو " "" "

الخطـــأ رقم 126 يعني ان الرمز المتوقع ظهوره هو برنامـــج

الخطـــأ رقم 127 يعني ان الرمز المتوقع ظهوره هو متغيـــر

الخطـــأ رقم 128 يعني ان الرمز المتوقع ظهوره هو برنـمـــج

الخطـــأ رقم 129 يعني ان الرمز المتوقع ظهوره هو مصفوفـــة

الخطـــأ رقم 130 يعني ان الرمز المتوقع ظهوره هو نـــوع

الخطـــأ رقم 131 يعني ان الرمز المتوقع ظهوره هو ابـــداء

الخطـــأ رقم 132 يعني ان الرمز المتوقع ظهوره هو انهـــي

الخطـــأ رقم 133 يعني ان الرمز المتوقع ظهوره هو اذا

الخطـــأ رقم 134 يعني ان الرمز المتوقع ظهوره هو عندئـــذ

الخطـــأ رقم 135 يعني ان الرمز المتوقع ظهوره هو بخلافـــ

الخطأ رقم 136 يعني ان الرمز المتوقع ظهوره هو بينما

الخطأ رقم 137 يعني ان الرمز المتوقع ظهوره هو نفذ

الخطأ رقم 138 يعني ان الرمز المتوقع ظهوره هو اقراء

الخطأ رقم 139 يعني ان الرمز المتوقع ظهوره هو أكتب

الخطأ رقم 140 يعني ان الرمز المتوقع ظهوره هو غير معرّف

الخطأ رقم 141 يعني ان الرمز المتوقع ظهوره هو لكل

الخطأ رقم 142 يعني ان الرمز المتوقع ظهوره هو الى

الارقام التالية تمثل ارقام وجمل الاخطاء التي يعلنها محلل المعنى

الخطأ رقم 143 يعني ان المتغير معرف مرتين

الخطأ رقم 144 يعني ان المتغير غير معرف

الخطأ رقم 145 يعني وجود عدم تطابق بين المتغير والتعريف الخاص به

الخطأ رقم 146 يعني ان المتغير المفهرس يجب ان يكون من نوع مصفوفة

الخطأ رقم 147 يعني ان فهرس التعبير يجب ان يكون من نوع صحيح

الخطأ رقم 148 يعني ان الحد يجب ان يكون من نوع منطق

الخطأ رقم 149 يعني ان الحد يجب ان يكون من نوع صحيح

الخطأ رقم 150 يعني ان كلا الحدين يجب ان تكون نوع صحيح او حرف

الخطأ رقم 151 يعني ان التعبير يجب ان يكون من نفس نوع المتغير

الخطأ رقم 152 يعني ان المتغير المقروء يجب ان يكون صحيح او حرف

الخطأ رقم 153 يعني ان القيمة المكتوبة يجب ان تكون صحيح او حرف

الخطأ رقم 154 يعني ان التعبير يجب ان يكون من نوع منطق