الحاسبات الالكترونية

مجلة علمية محكمة تعنى بعلوم الحاسبات الالكترونية

(نصف سنوية)

عدد خاص

بحوث المؤتمر الوطني الرابع للحاسبات الالكترونية

٢٥ - ٢٧ / تشرين ثاني / ١٩٩٦

الجهات المنظمة

الجمعية العراقية لعلوم الحاسبات والمركز القومي للحاسبات الالكترونية

وكلية الرافدين الجامعة واتحاد مجالس البحث العلمي العربية

العدد الثامن والعشرون - السنة الثامنة عشر / ١٩٩٦

تصدر عن : وزارة التعليم العالي والبحث العلمي / المركز القومي للحاسبات الالكترونية

الحاسبات الالكترونية

مجلة علمية محكمة تعنى بعلوم الحاسبات الالكترونية

(نصف سنوية)

العدد الثامن والعشرون - السنة الثامنة عشر / ١٩٩٦

تصدر عن : وزارة التعليم العالي والبحث العلمي / المركز القومي للحاسبات الالكترونية

الدعم البرمجي للتطبيقات الحديثة على الحاسبات الالكترونية حالة دراسية : رسم نص القران الكريم على جدران احدى مساجد بغداد بواسطة الحاسبة

المهندس خالد محمد خالد

العراق - بغداد - الاعظمية - م 318 ش 27 د 71 ت 4227715

المستخلص

على الرغم من كفاءة التطبيقات الحديثة للحاسبة الالكترونية وكثرة الادوات التي توفرها ، الا انها لايمكن ان تناسب جميع الاحتياجات المتخصصة . ومن جانب اخر فان الخبرة البرمجية للافراد مهما تطورت فانها لاترتقي للوصول الى امكانية تطوير تطبيقات بكفاءة التجارية الجاهزة لكل حاجة تخصصية ويهدف هذا البحث الى توضيح امكانية التوظيف المزدوج للامكانيات المتطورة للتطبيقات الحديثة مع التكيف البرمجي لها للاحتياج المتخصص والذي يمثل افضل طريقة لانجاز المهام المعقدة . فان تطويع قدرات البرامج الحديثة عبر الدعم البرمجي باستخدام اللغات البرمجية الشائعة او لغات البرمجة الخاصة بالتطبيق يضاعف كفاءة التطبيقات بمرات عديدة .

ويتطرق البحث لوصف وتقييم الدعم البرمجي للتطبيقات الحديثة على الحاسبة الالكترونية بدراسة حالة تطبيقية وهي تنفيذ رسم النص القرآني الكامل على جدران احدى المساجد في مدينة بغداد . وقد نتج عن طريق استخدام الحاسبة مع الدعم البرمجي لتنفيذ المشروع اختصار جهد سنوات من العمل الى اقل من 170 ساعة عمل فعلية. والبحث يستعرض تفاصيل المهمة مع التركيز على الجهد البرمجي المكمل لاداء البرامج التجارية الجاهزة من حيث تهيئة البيانات وتحويلها وتسيير البرامج الجاهزة للعمل الذاتي للاسراع بتنفيذ المهمة وتقليل الجهد البشري في انتاج العمل مما يودي الى تقليل الاخطاء كما يستعرض البحث تفاصيل خوارزمية التعريب المستخدمة في اظهار النص القرآني الكريم بالحرف الكوفي المختار لتنفيذ المهمة .

الدعم البرمجي للتطبيقات الحديثة على الحاسبات الالكترونية
حالة دراسية : رسم نص القرآن الكريم على جدران إحدى مساجد بغداد بواسطة الحاسبة

المهندس خالد محمد خالد
العراق - بغداد - الأعظمية - م ٣١٨ ش ٢٧ د ٧١ ت ٤٢٢٧٧١٥

وصف المهمة

كان المطلوب هو رسم النص الكامل للقرآن الكريم ضمن لوحات جدارية ذات تصميم معين. وهذه اللوحات مرتبة بعدد محدد وبأبعاد ٤،٦٠م × ١م للوحة الواحدة . وتم اختيار الحرف الكوفي وطلب تحديد الحجم المناسب للحرف بما يسهل قراءته خاصة عند أعلى اللوحات . وبهذا فقد تحددت المسألة الرئيسية المتبقية في اختيار الحجم المناسب للحرف بما يمكن كتابة النص الكامل للقرآن الكريم ضمن المساحة المحددة ودون ترك فراغات فيها . وتم تحديد المهام التالية لإنجاز العمل :

١- تحويل تصميم طقم الحروف الكوفية الى الحاسبة
٢- تهيئة النص القرآني الكريم .
٣- إجراء الحسابات لتحديد حجم الحرف .
٤- توزيع النص على اللوحات وبالشكل المطلوب .
٥- تسقيط الملفات المقسمة على اللوحات.

الأجهزة والأدوات البرمجية التي اختيرت لإنجاز المهمة

تقرر تنفيذ المهمة على حاسبات متوافقة مع الـ IBM-PC لتوفرها مع وجود المعرفة في استخدامها واختير برنامج اوتوكاد AutoCAD لإعداد الرسوم النهائية لإمكانيته في تحريك الراسمات وإنتاج رسوم بحجم (A0) إضافة الى التسهيلات التي يوفرها البرنامج في إضافة أدوات برمجية تمكننا من تنفيذ متطلباتنا الخاصة عبر لغة AutoLISP المدمجة مع أوتوكاد . وتطلب تحويل رسومات الحروف الكوفية من الورق إستخدام جهاز Scanner منضدي بحجم A4 مع برامجه لتصوير الحروف وتحويلها الى ملفات على قرص الحاسبة. أما تطوير البرامج التحويلية ومولدات المعلومات فقدتم عن طريق إستخدام لغة C وبيئة برمجة Borland C المتكاملة.

تحويل تصميم طقم الحروف الكوفية المصممة الى الحاسبة

تم تزويدنا بتصميم طقم الحروف على الورق فتم تصوير الحروف باستخدام الـ Scanner ومن ثم تم تحويل الصور عن طريق برامج الـ Tracing الى رسومات خطية من نوع Vector ، حيث تم تحسين أشكال الحروف ببرنامج اوتوكاد وخزن كل حرف بشكل بلوك (Block) . وقد اختيرت طريقة تمثيل الحروف بشكل بلوك للمرونة التي تعطيها هذه الطريقة في معالجة الأسطر لاحقاً . ذلك لان الطريقة التي اعتمدت في تثبيت النص على الأسطر هي طريقة الفراغ المتغير . أي أن الفراغ بين الكلمات يتغير لاستيعاب النص ضمن الأسطر وبسيطرة برمجية من قبلنا . هذا وتم كتابة ملف منفصل لوصف أبعاد كل حرف وذلك للاستخدام في الحسابات اللاحقة لاستيعاب النص .

تهيئة النص القرآني

تم إيلاء تهيئة النص القرآني أهمية بالغة لحساسية الموضوع وعدم احتماله للخطأ . وقد حاولنا تحاشي تنضيد النص لما يتطلبه ذلك من الوقت والجهد في التنسيق . وجرى البحث عن إمكانية الحصول على نص مدقق ومخزون على أجهزة الحاسبة . ومن البرامج التي حصلنا عليها هو برنامج سلسبيل للقرآن الكريم . وقد تم خزن النص القرآني في هذا البرنامج بطريقة خاصة في عشرة ملفات تقل من حجم الخزن الى (100 KB) . وكان من المتعذر الحصول على النص من هذه الملفات مباشرة . إلا أن برنامج سلسبيل يرسل النص القرآني الى الطابعة . فقمنا بتحويل مسار الإخراج الى ملف على القرص بدلاً من الطابعة بواسطة برنامج خاص لذلك ومن ثم كتابة برنامج لتحويل تجفير الحروف من طريقة سلسبيل الى إحدى التجفيرات الشائعة والتي اخترنا لها تجفير (نافذة) Nafitha Code Page. وكان الناتج هو ملف بحجم (700 KB) تقريباً بتجفير نافذة يمكن التعامل معه بشكل مباشر ودون توسط برنامج سلسبيل .

تنزيل النص على اللوحات الجدارية المحددة

تطلب توزيع النص على اللوحات الجدارية أجراء العمليات التالية :

أولاً احتساب حجم الحرف المناسب ومن ناحيتين :

١- إمكانية استيعاب النص كاملاً .

٢- سهولة قراءة الحرف وعلى الارتفاع المطلوب (٤،٦م تقريباً) .

وقد تم كتابة برنامج خاص لتحميل مواصفات الحروف ومن ثم القيام بعمليات حسابية بعد قراءة ملف النص القرآني لاحتساب عدد السطور بموجب تحديد عرض الفراغ التقريبي بين الكلمات . وقد تطلب الأمر عدة جولات من الاحتساب والتغيير حتى تم الوصول الى الحجم المناسب للحرف . ومن ثم تم تقديم نموذج الى الجهة المستفيدة لتأييد استخدام الحجم المختار من حيث مناسبته للقراءة على الارتفاع المطلوب.

ثانياً تقسيم النص القرآني الى ملفات : وتمت هذه العملية بعد تحديد حجم ورقة الرسم (والتي استخدمت كقالب للتنفيذ في النهاية) وتقسيم تصميم اللوحة الجدارية على ٦ ورقات . ولم يكن تصميم اللوحة الجدارية متشابهاً في كل أجزائه حيث تضمن تخصرات تصميمية في الأعلى والأسفل (شكل رقم ٣) ولذا فقد تم ترميز ورقات اللوحات وكتابة ملف للتصميم يتضمن وصفاً لكل سطر من أسطر اللوحات من حيث العرض . ومن ثم تم كتابة برنامج خاص لقراءة الملف التصميمي والنص القرآني حيث يقوم البرنامج بكتابته النص القرآني الى ملفات يتضمن كل ملف جزء النص الخاص بلوحة محددة . ويقوم البرنامج بتسمية الملفات ذاتياً وحسب ترميز اللوحات المتبع . وقد تم اختيار هذه الطريقة لرونتها في إمكانية التحقق من استيعاب كل لوحة للنص بشكل كامل ومتجانس مع إمكانية أجراء التصحيحات والرتوش اليدوية الأخيرة قبل البدء في تحويل ملفات النص الى لوحات رسم بواسطة برنامج اوتوكاد . وقد أثبتت الإجراءات الحسابية في المرحلة الأولى جدواها لتقليل جهد التقسيم[سرد برمجة ١ في الملحق].

ثالثاً : تسقيط ملفات النص وتوليد الرسومات : وتمت هذه المرحلة كلياً داخل برنامج اوتوكاد . وتم إعداد برامج بلغة AutoLISP مع ملفات Script خاصة لتحويل ملفات النص المجزأ الى الرسومات المطلوبة وبطريقة ذاتية بالكامل ، ذلك لان حجم العمل من حيث عدد الرسومات والتفاصيل الداخلة في كل رسم يمثل جهداً لا يمكن تحقيقه ضمن المدة

المحددة إلا باستخدام الحاسبة بإمكانيات برمجية مناسبة .

وقلب هذه العملية يتمثل ببرنامج مكتوب بلغة AutoLISP يقوم بقراءة ملفات النص المجزأ وتسقيط الأسطر باستخدام الحرف المصمم وضمن الإطار التصميمي للوحة ، مع وضع رموز فواصل الآيات وترقيمها وبدايات السور وعناوينها . ويعتمد هذا البرنامج على طريقة تغيير الفراغ بين الكلمات لتثبيت السطور ضمن العرض المحدد . وهي الطريقة المستخدمة في بعض برامج النشر المكتبي . ولم يتم اعتماد طريقة إطالة الكلمات لان الطريقة الأولى تعطي نسيجاً متوازناً تشكل النص وتحافظ على جمال الكلمات المفردة . وتبرز مشكلة السطر الأخير من كل سورة ، حيث أن النص قد لا يكفي للئ السطور . وهنا يقوم البرنامج بالتقدير الذاتي حيث انه لا يقوم بتعديل الفراغ ويثبت النص في منتصف السطر وبشكل متوازن . وبعد الانتهاء من التسقيط يقوم البرنامج بترقيم الرسم في مكان معين على الجانب ثم يخزن الرسم ذاتياً . وفي الواقع فإن التحميل والخزن لم يتم عبر برنامج AutoLISP المذكور وذلك لان نسخة اوتوكاد r10 التي استخدمت لتنفيذ المشروع لم تكن تتضمن إمكانية فتح رسوم جديدة وخزنها بلغة AutoLISP [سرد برمجة ٢ في الملحق]. لذا استخدمت ملفات SCRIPT لتوجيه اوتوكاد الى فتح رسم جديد باسم معين وتشغيل برنامج الـ AutoLISP لتنفيذ الرسم ثم وبعد الانتهاء من الرسم خزنه وفتح ملف جديد وهكذا حتى نهاية التسقيط . وقد كتب برنامج خاص بلغة C لتوليد ملفات SCRIPT ذاتياً وذلك للتقليل من الأخطاء في تسمية الملفات وترميزها. وقد مكننا هذه الإجراءات من إختصار فترة الإنتاج بنسبة عالية جدا.

خوارزمية التعريب

ان إظهار الحروف العربية على الشاشة أو عند الطباعة على الورق يتطلب اجراء عملية تعريبية معينة تتعلق بربط الحروف وتسمى التحليل الموضعي Contextual Analysis . فالحرف العربي الواحد يمكن أن يظهر في الكتابة او الطبع بعدة أشكال . فحرف الباء مثلاً يمكن ان يظهر بأربعة أشكال : أولى ، وسطي ، نهائي ، ومنفصل (بـ ـبـ ـبـ ب) . هذا من جانب ، ومن جانب آخر فان بعض الحروف لا يرتبط مع الحروف التي تليها كالالف والواو والدال . كما ان النص قد يحتوي على الأرقام والرموز التي لا تحتاج الى الربط اساساً .

ان الكتابة اليدوية او الطباعة على الآلة الكاتبة الميكانيكية والوسائل التنضيدية القديمة تعتمد على الجهد البشري في تحديد شكل الحرف حسب موقعه من الكلمة والسطر . ومع دخول الحاسبة الالكترونية أصبح بالإمكان إجراء الربط الذاتي للحرف ، حيث ان الحاسبة هي التي تقوم بربط الحروف اعتماداً على خوارزمية التعريب .

وحينما تم تحويل النص القرآني لمشروعنا من الناتج الطباعي لبرنامج سلسبيل الى ملف بتجفير نافذة (Nafitha Code Page) تم إزالة التحليل الموضعي ، بمعنى ان الحرف الأولى والوسطي والنهائي والمنفصل لتجفير سلسبيل الطباعي حولت الى الرمز المقابل في تجفير نافذة كحرف مجرد ، وذلك لاسباب عديدة منها ان نظام تعريب نافذة الذي استخدم لتعريب برامج معالجات النصوص لتعديل النص هو الذي يوفر ربط الحروف بإجراءاته الخاصة للتحليل الموضعي . والشيء الثاني والاهم هو ان متطلبات التعريب للحرف المستخدم في إنتاج الرسوم النهائية تختلف عن متطلبات حرف سلسبيل الطباعي . لذا تم إجراء

عملية التحليل الوضعي المناسب للحرف المستخدم في برامج التحويل التي قمنا بكتابتها .

وبنيت خوارزمية التحليل الوضعي التي اعتمدناها على طريقة استخدام جدول بحث Lookup Table (LUT) والذي يعتمد في بحثه على تقسيم المجموعة الطباعية الى مجموعة الأحرف فقط ومنها مجموعة الأحرف المرتبطة فقط . وعولج النص باسلوب خطي ابتداءاً من أول النص وحتى آخره وفي مرة واحدة [سرد برمجة ٣ في الملحق].وتتضمن الخوارزمية الخطوات التالية :

١- قراءة حرف النص الغير محول من ملف الإدخال المجفر بتجفير نافذة .

٢- قراءة الحرف المحول السابق ان وجد .

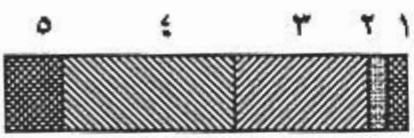
٣- تحديد شكل الحرف بناء على القراءتين .

٤- استخدام دالة الاستبدال لاختيار شكل الحرف .

٥- إرسال الحرف المستبدل الى الإخراج .

شكل رقم ١

مقارنة نسبية للمراحل التنفيذية



١- تصوير الحروف
٢- تحويل الصور الى Vector DXF
٣- تحسين الحروف وإعادة الرسم
٤- العمل البرمجي والتحويلي
٥- توليد الرسومات

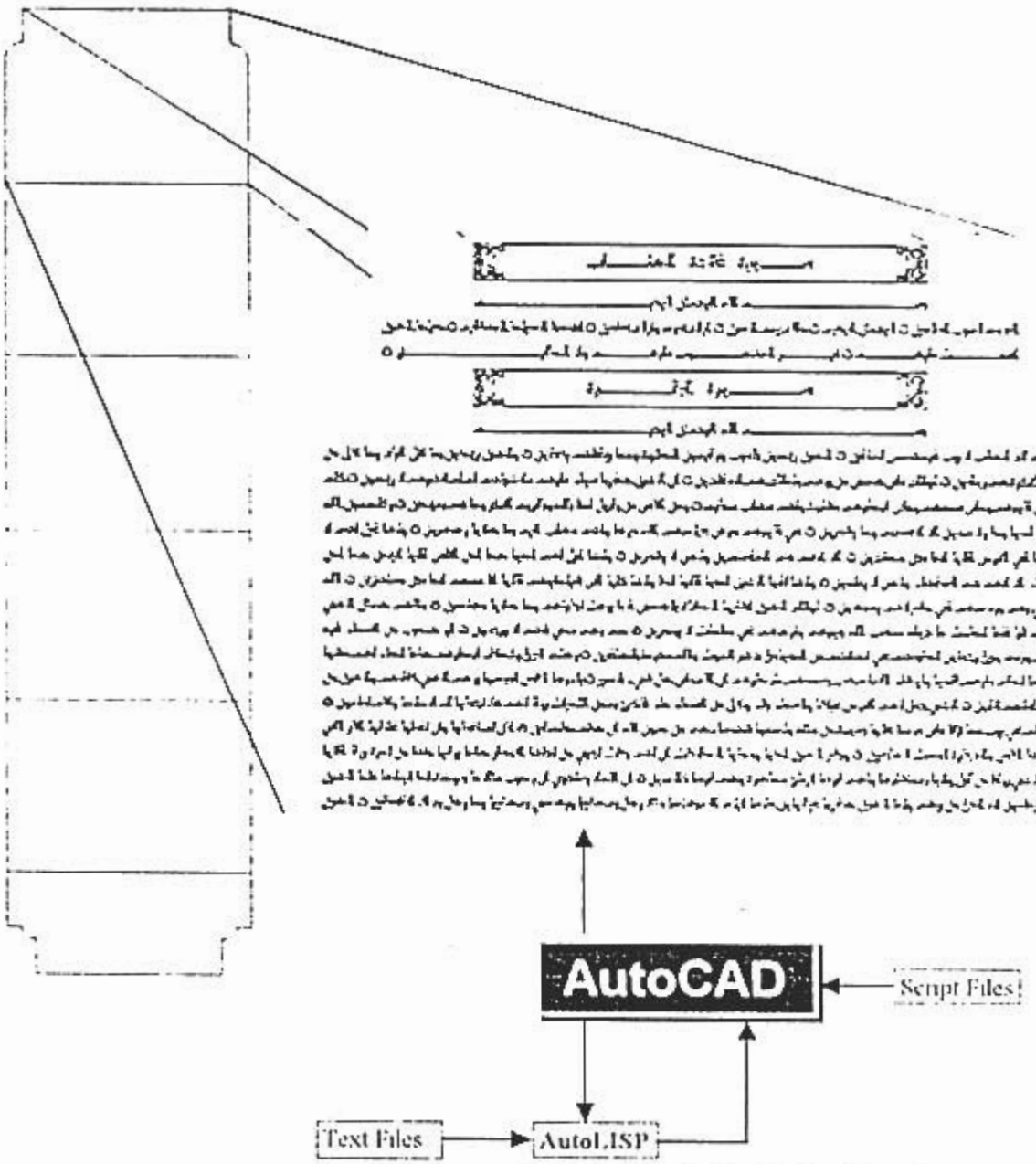
شكل رقم ٢

نسبة التسريع باستخدام الحاسبة

References
[1] AutoCAD r10, Autodesk Inc., Copyright 1982 - 1988.
[2] AutoLISP r10, Autodesk Inc., Copyright 1982 - 1988.
[3] Borland C, ver. 2.0, Borland International Inc., Copyright 1990 - 1991.
[4] Capture, Insight Development Co., Copyright 1987-1988.
[5] Herbert Schildt, The Complete C Reference, McGraw Hill, 1990.

شكل رقم ٣

تصميم اللوحة الجدارية (على اليسار) ومخطط الوحدات البرمجية لتوليد الرسومات



شكل رقم ٤

نموذج الحرف المستخدم بالحجم الطبيعي

فقالوا انا سمعنا قرانا عجبا ۝ ١

سورة فاتحة الكتاب

بسم الله الرحمن الرحيم

الحمد لله رب العالمين ۝ الرحمن الرحيم ۝ مالك يوم الدين ۝ اياك نعبد واياك نستعين ۝ اهدنا الصراط المستقيم ۝ صراط الذين
انعمت عليهم ۝ غير المغضوب عليهم ولا الضالين ۝

سورة البقرة

بسم الله الرحمن الرحيم

الم ۝ ذلك الكتاب لا ريب فيه هدى للمتقين ۝ الذين يؤمنون بالغيب ويقيمون الصلوة ومما رزقناهم ينفقون ۝ والذين يؤمنون بما انزل اليك وما انزل من
قبلك وبالاخرة هم يوقنون ۝ اولئك على هدى من ربهم واولئك هم المفلحون ۝ ان الذين كفروا سواء عليهم ءانذرتهم ام لم تنذرهم لا يؤمنون ۝ ختم
الله على قلوبهم وعلى سمعهم وعلى ابصارهم غشاوة ولهم عذاب عظيم ۝ ومن الناس من يقول امنا بالله وباليوم الاخر وما هم بمؤمنين ۝ يخادعون الله
والذين امنوا وما يخدعون الا انفسهم وما يشعرون ۝ في قلوبهم مرض فزادهم الله مرضا ولهم عذاب اليم بما كانوا يكذبون ۝ واذا قيل لهم لا
تفسدوا في الارض قالوا انما نحن مصلحون ۝ الا انهم هم المفسدون ولكن لا يشعرون ۝ واذا قيل لهم امنوا كما امن الناس قالوا انؤمن كما امن
السفهاء الا انهم هم السفهاء ولكن لا يعلمون ۝ واذا لقوا الذين امنوا قالوا امنا واذا خلوا الى شياطينهم قالوا انا معكم انما نحن مستهزءون ۝ الله
يستهزئ بهم ويمدهم في طغيانهم يعمهون ۝ اولئك الذين اشتروا الضلالة بالهدى فما ربحت تجارتهم وما كانوا مهتدين ۝ مثلهم كمثل الذي
استوقد نارا فلما اضاءت ما حوله ذهب الله بنورهم وتركهم في ظلمات لا يبصرون ۝ صم بكم عمي فهم لا يرجعون ۝ او كصيب من السماء فيه
ظلمات ورعد وبرق يجعلون اصابعهم في اذانهم من الصواعق حذر الموت والله محيط بالكافرين ۝ يكاد البرق يخطف ابصارهم كلما اضاء لهم مشوا
فيه واذا اظلم عليهم قاموا ولو شاء الله لذهب بسمعهم وابصارهم ان الله على كل شيء قدير ۝ يا ايها الناس اعبدوا ربكم الذي خلقكم والذين من
قبلكم لعلكم تتقون ۝ الذي جعل لكم الارض فراشا والسماء بناء وانزل من السماء ماء فاخرج به من الثمرات رزقا لكم فلا تجعلوا لله اندادا وانتم تعلمون ۝
وان كنتم في ريب مما نزلنا على عبدنا فاتوا بسورة من مثله وادعوا شهداءكم من دون الله ان كنتم صادقين ۝ فان لم تفعلوا ولن تفعلوا فاتقوا النار التي
وقودها الناس والحجارة اعدت للكافرين ۝ وبشر الذين امنوا وعملوا الصالحات ان لهم جنات تجري من تحتها الانهار كلما رزقوا منها من ثمرة رزقا قالوا
هذا الذي رزقنا من قبل واتوا به متشابها ولهم فيها ازواج مطهرة وهم فيها خالدون ۝ ان الله لا يستحيي ان يضرب مثلا ما بعوضة فما فوقها فاما الذين
امنوا فيعلمون انه الحق من ربهم واما الذين كفروا فيقولون ماذا اراد الله بهذا مثلا يضل به كثيرا ويهدي به كثيرا وما يضل به الا الفاسقين ۝ الذين

الملحق (نماذج من السرد البرمجي)

سرد برمجة ١

```
// File Partitioning Based on a Previously Calculated Text Scale
void partition(char *infilename, char *outfilename)
{ int line_length = 0, i = 0, current_line_design;
  FILE *input_stream, *output_stream;

        input_stream = fopen(infilename, "r");
        output_stream = fopen(outfilename,"w");
        current_line_design = nextLineDesign();
        while(!feof(input_stream)){
                ch = getch(input_stream);
                line_length += getcharWidth(ch);
                        if(line_length >= current_line_design){
                                line[i] = 0; // Terminate the string
                                fputs(line, output_stream);
                                i = 0; line_length = 0;
                                current_line_design = nextLineDesign();
                        }
                        else line[i++] = ch;
        }
        if(i > 0) fputs(line, output_stream);
        fclose(input_stream); fclose(output_stream);
}
```

سرد برمجة ٢

```
; AutoLISP Panel Plotting Routine
(defun fit_line(str / textlen designlen spacenum space)
        (setq   text_len (gettextlen str scale)
                spacenum (1- (getwordnum str)
                designlen (getdesignlen scale)
        )
        (if (< textlen (* 0.8 designlen))
         (setq space defaultspace justification "m" suraendflag t)
         ;else
         (setq space (/ (- designlen textlen)(float spasenum)) justification "f")
        )
        (draw_line str space justification)
)
```

سرد برمجة ٣

```
// Arabization Lookup Table
char alef[]             = {101,102,102,101};
char ba[]               = {103,104,105,106};
char ta[]               = {107,108,109,110};
char tha[]              = {111,112,113,114};
char jeem[]             = {115,116,117,118};
char hha[]              = {119,120,121,122};
char kha[]              = {123,124,125,126};
char dal[]              = {127,128,128,127};
char dhal[]             = {129,130,130,129};
char ra[]               = {131,132,132,131};
char za[]               = {133,134,134,133};
char seen[]             = {135,136,137,138};
char sheen[]            = {139,140,141,142};
char sad[]              = {143,144,145,146};
char dhad[]             = {147,148,149,150};
char tta[]              = {151,152,153,154};
char ttha[]             = {155,156,157,158};
char ain[]              = {159,160,161,162};
char ghain[]            = {163,164,164,166};
char fa[]               = {167,168,169,170};
char quaf[]             = {171,172,173,174};
char kaf[]              = {175,176,177,178};
char lam[]              = {179,180,181,182};
char meem[]             = {183,184,185,186};
char noon[]             = {187,188,189,190};
char ha[]               = {191,192,193,194};
char waw[]              = {195,196,196,195};
char ya[]               = {197,198,199,200};
char marboota[]         = {201,202,202,201};
```

تتمة سرد برمجة ٣

```
char kursi[]            = {203,204,205,206};
char hamzwaw[]          = {207,208,209,207};
char lamalef[]          = {209,210,210,209};
char maqsura[]          = {216,215,215,216};
char lamalefuhamz[]     = {213,214,214,213};
char lamaleflhamz[]     = {219,220,220,219};
char aleflhamz[]        = {217,218,218,217};
char alefwasl[]         = {221,222,222,221};

// Character groups based on the Context Analysis classification
char lett[] =
 166,167,169,168,170,171,172,173,224,225,226,227,228,229,230,231,232,233,234,235,236,2
37,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253};


// Concatenating letter set
char precat[] =
 170,172,224,225,226,227,228,233,234,235,236,237,238,239,240,242,243,244,245,246,248,2
49,253};


void context_analysis(char *str1, char *str2)
{       char buffer[256];
        int i, len, pre,ch , post, position, direct;

        len = strlen(str1);
        for(i = 0; i <= len; i++){
                if(i == len) post = 0; else post = str1[i + 1];
                direct = FALSE;
                if(!in(pre, precat) && in(ch, lett) && in(post, lett)) position = 0;
                else if(in(pre, precat)&& in(ch, lett)&& in(post, lett  position = 1;
                else if(in(pre, precat)&& in(ch, lett)&& !in(post, lett)) position = 2;
                else if(!in(pre, precat) && in(ch, lett  && !in(post, lett
                                                                position = 3;
                else direct = TRUE;

                if(direct){ buffer[i] = ch; direct = FALSE;}
                else buffer[i] = replace(ch, position );
        }
        buffer[i] = '\0';
        strcpy(str2, buffer);

}

char replace(char ch, int position)
{

        switch(ch){
                case 167: return alefuhamz[position];
                case 169: return aleflhamz[position];
                case 171: return alef[position];
                case 172: return ba[position];
                case 224: return ta[position];
                case 225: return tha[position];
                case 226: return jeem[position];
                case 227: return hha[position];
                case 228: return kha[position];
                case 229: return dal[position];
                case 230: return dhal[position];
                case 231: return ra[position];
                case 232: return za[position];
                case 233: return seen[position];
                case 234: return sheen[position];
                case 235: return sad[position];
                case 236: return dhad[position];
                case 237: return tta[position];
                case 238: return ttha[position];
                case 239: return ain[position];
                case 240: return ghain[position];
                case 242: return fa[position];
                case 243: return quaf[position];
                case 244: return kaf[position];
                case 245: return lam[position];
                case 246: return meem[position];
                case 248: return noon[position];
                case 249: return ha[position];
```

```
                case 251: return waw[position];
                case 253: return ya[position];
                case 173: return marboota[position];
                case 170: return kursi[position];
                case 166: return alefwas[position];
                case 168: return hamzwaw[position];
                case 250: return lamalef[position];
                case 252: return maqsura[position];
                case 241: return lamalefuhamz[position];
                case 247: return lamaleflhamz[position];
                default: return ch;
        }
}
```