



# تقرير مشروع المترجمات

## بيئة تطوير لغة البرمجة العربية

الجمهورية اليمنية

جامعة إب - كلية العلوم التطبيقية

قسم علوم الحاسوب وتقنية المعلومات

مترجمات

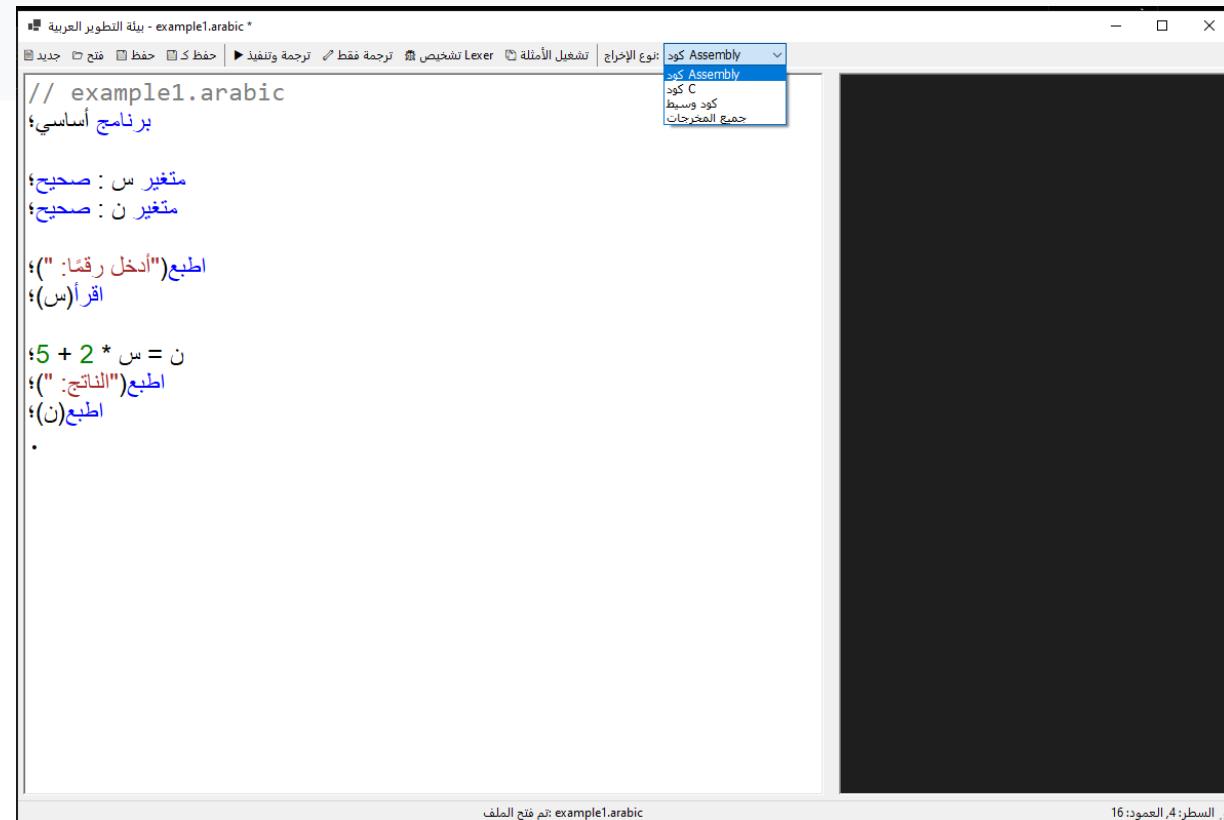
المشرف: د/ خالد الكحشه

الطلاب:

- أيمن محمد ناجي فمحان
- حازم هزام جمال العمري
- ضياء فضل الحضرمي
- طارق فضل علي محمد العمري
- علي محمد أحمد الفواز

# المقدمة والملخص

- المشروع: لغة برمجة عربية مع بيئة تطوير متكاملة (Compiler) ومتراجم (IDE). 
- الهدف: تمكين كتابة الكود بالعربية وتحويله إلى C ثم تنفيذ البرنامج. 
- المخرجات التعليمية: فهم خط أنابيب المترجم، بناء شجرة النحو المجرد (AST)، توليد كود وسيط ونهائي، وتصميم واجهة IDE عربية. 



# مكونات المشروع

## المترجم (++C)

التحليل اللغوي والنحوي للكود المصدري العربي ✓

بناء شجرة النحو المجرد (AST) ✓

توليد الكود الوسيط (IR) ✓

توليد الكود النهائي بلغة C ✓

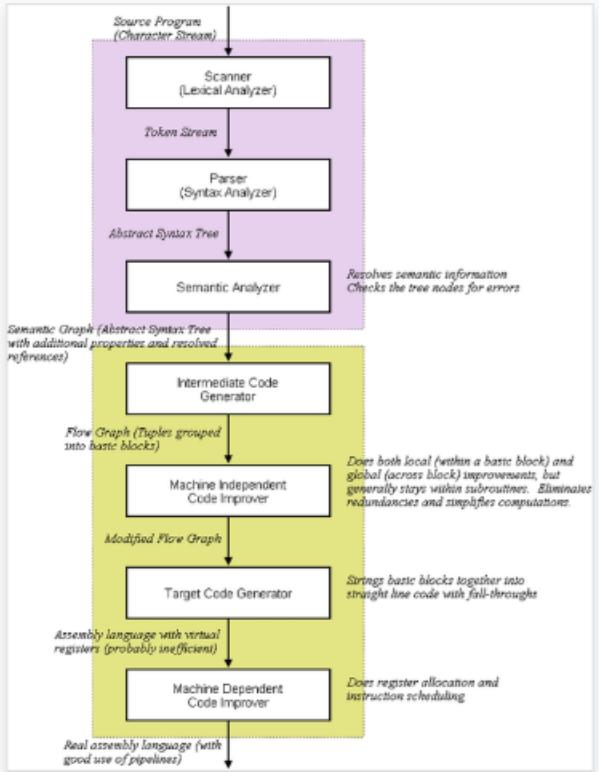
## بيئة التطوير (#C)

واجهة رسومية مبنية باستخدام Windows Forms.NET 6.0 و ✓

محرر نصي مع دعم كامل للغة العربية ✓

تنفيذ الكود واستعراض نتائج كل مرحلة ✓

ادارة ملفات وتكامل مع المترجم والأدوات الخارجية ✓



# بيئة التطوير المتكاملة (IDE) – الوظائف

## ادارة الملفات

فتح، حفظ، حفظ باسم، وإنشاء ملفات جديدة ✓

أختصارات: Ctrl+N، Ctrl+O، Ctrl+S 📁

## ✿ التشغيل والتتنفيذ

استدعاء المترجم → توليد كود C → استدعاء C ✓

تشغيل البرنامج وعرض المخرجات (F5) ✓

## ❶ عرض مراحل الترجمة

التحليل المعجمي - عرض الرموز (Tokens) ✓

التحليل النحوي - عرض الشجرة (AST) ✓

عرض الكود الوسيط (IR) والكود النهائي (C) ✓

The screenshot shows the IDE interface with the following components:

- Code Editor:** Displays the source code in Arabic. The code includes variable declarations like "ناتج", arithmetic operations like "ناتج = رقم1 + رقم2;", and print statements like "اطبع ناتج;".
- Terminal Window:** Shows the output of the compiler. It includes the command line, file paths for temporary files, assembly code, and object code.
- Status Bar:** Shows the date and time (AM 14:12, 28/10/2025), and a message in Arabic: "الترجمة تمت بنجاح" (Translation completed successfully).

# المرحلة 1: التحليل المعجمي (Lexical Analysis)

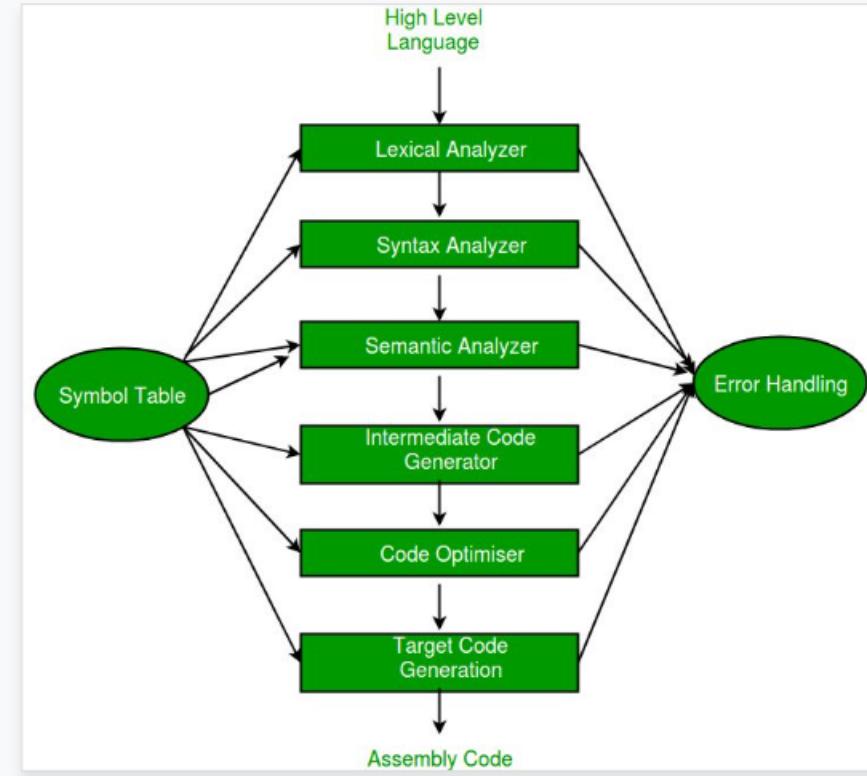
## ما هو التحليل المعجمي؟

المرحلة الأولى في عملية الترجمة التي تقوم بقراءة الكود المصدرى كتيار من الحروف وتقسيمه إلى وحدات أولية ✓

تحويل النص إلى رموز مميزة (Tokens) مثل: كلمات محجوزة، معرفات، ثوابت، مشغلات ✓

استبعاد التعليقات والمسافات غير الضرورية في هذه المرحلة ✓

## أمثلة على الرموز (Tokens) </>



"متغير" - TOKEN\_KEYWORD\_VAR

"سر" - TOKEN\_IDENTIFIER

"=" - TOKEN\_OPERATOR

"TOKEN\_NUMBER" - "10"

"IF" - TOKEN\_KEYWORD\_IF

## المرحلة 2: التحليل النحوی وبناء AST

### التحليل النحوی (Syntax Analysis)

تحقق من القواعد النحوية للغة البرمجة العربية ✓

تحويل قائمة الرموز (Tokens) إلى هيكل شجري ✓

اكتشاف الأخطاء النحوية وتقديم رسائل دقيقة ✓

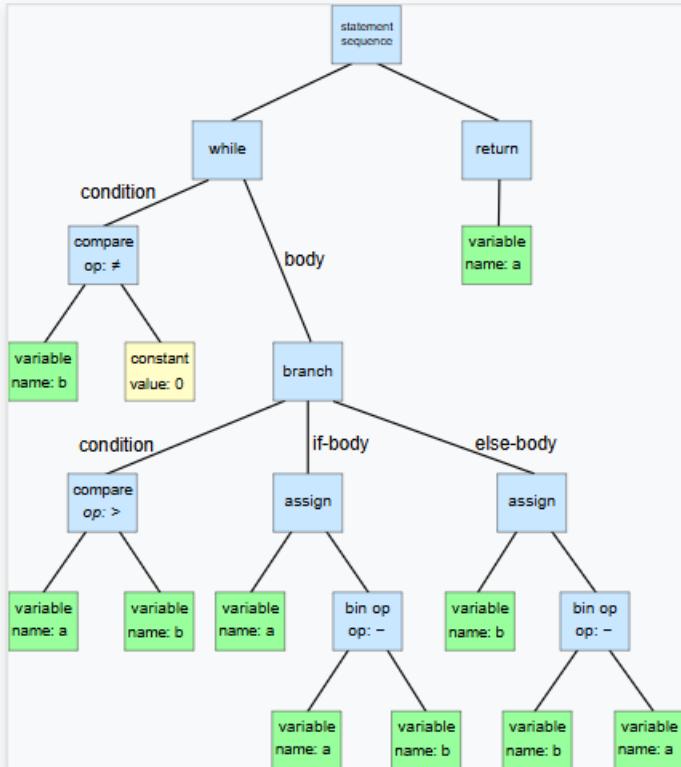
### هيكل شجرة النحو المجرد (AST)

تمثيل بنية البرنامج بشكل هرمي ومنطقي ✓

العقد تمثل العمليات والمتغيرات والثوابت والتعبيرات ✓

تساعد في توليد الكود الوسيط بشكل فعال ✓

تحفظ بالترتيب المنطقي للكود بدون تفاصيل النحو الزائدة ✓



### المرحلة 3: توليد الكود الوسيط (IR)

#### </مثال للكود الوسيط (Three-Address Code)>

The screenshot shows a software interface for generating intermediate code (Three-Address Code). The left pane displays the source code in Arabic:

```
// example1.arabic
برنامِج أساسی؛

متغیر س : صحیح;
متغیر ن : صحیح;

اطبع("ادخل رقمًا: ");
اقرأ(س);

ن = س * 5 + 2 ;
اطبع("الناتج: ");
اطبع(ن);
```

The right pane shows the generated Three-Address Code:

```
0: PRINT STRING, str_0
1: READ س
2: LOAD t0, س
3: LOAD t1, 2
4: MUL t2, t0, t1
5: LOAD t3, 5
6: ADD t4, t2, t3
7: STORE ن, t4
8: PRINT STRING, str_1
9: PRINT VARIABLE, ن
10: HALT
```

Below the code, there are sections for statistics and file paths:

- إحصائيات:
  - عدد التعليمات: 11
  - عدد المتغيرات المؤقتة: 5
  - عدد العلامات: 0
  - عدد السلالس التصورية: 2
  - عدد الرموز في جدول الرموز: 2
  - الترجمة تمت بنجاح
- الملفات الناتجة:
  - C:\Users\USER\AppData\Local\Temp\tmpB763\_output\_intermed
  - C:\Users\USER\AppData\Local\Temp\tmpB763\_output.c (كود C)
  - C:\Users\USER\AppData\Local\Temp\tmpB763\_output.asm (كود Assembly)

At the bottom, a message indicates the translation was successful:

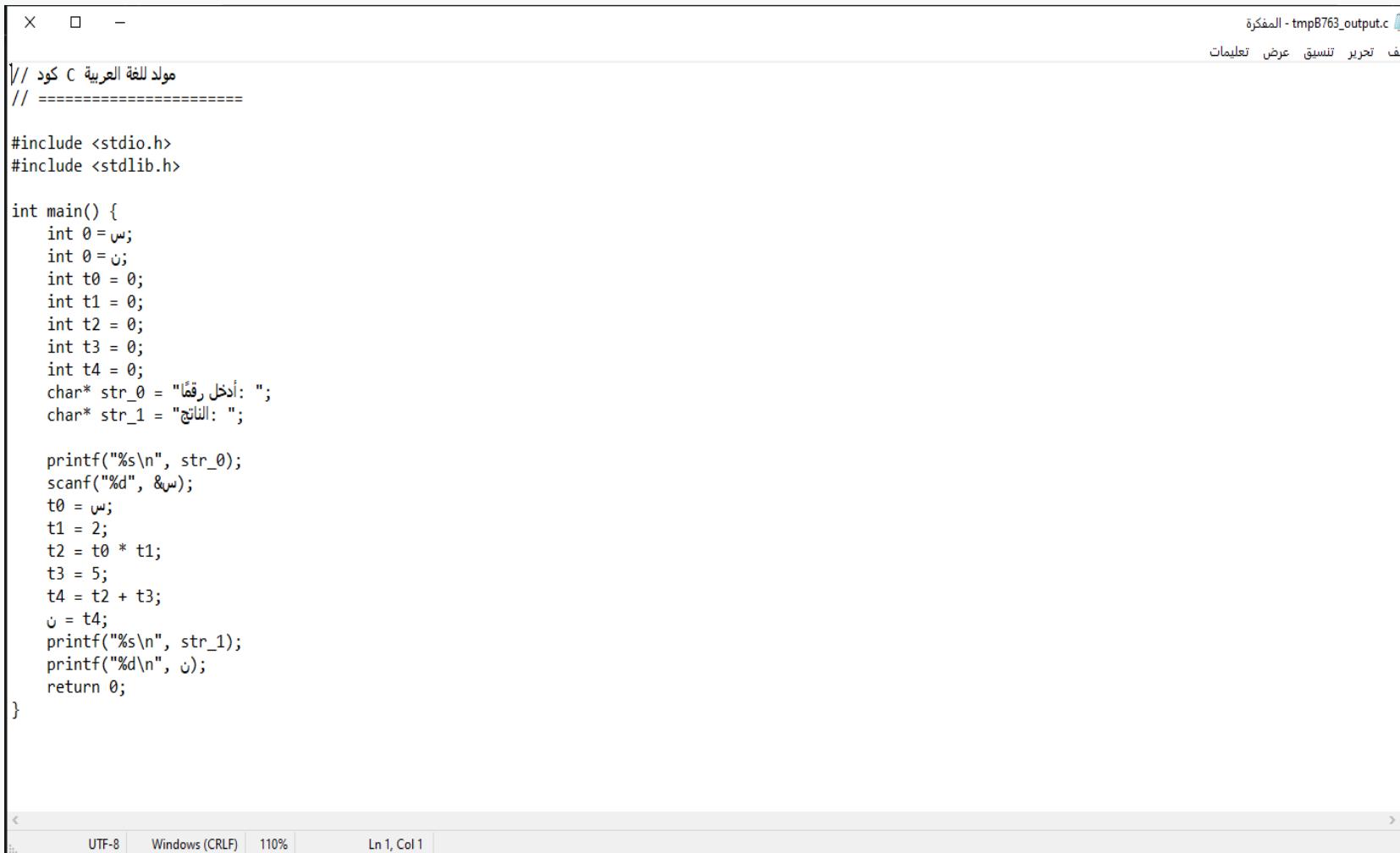
الترجمة تمت بنجاح

السطر: 1, العمود: 1

يمثل الكود الوسيط صفيلاً داخلياً أبسط للبرمجة وأكثر استقلالية عن اللغة المصدر. سيتم استخدامه لاحقاً لتوليد الكود النهائي أو إجراء تحسينات اختيارية.

## المرحلة 4: توليد الكود النهائي (C Transpilation)

</> مقتطف كود C المولد



The screenshot shows a code editor window with the following C code:

```
// مولد لغة العربية C كود
// =====

#include <stdio.h>
#include <stdlib.h>

int main() {
    int س = 0;
    int ن = 0;
    int t0 = 0;
    int t1 = 0;
    int t2 = 0;
    int t3 = 0;
    int t4 = 0;
    char* str_0 = "أدخل رقمًا: ";
    char* str_1 = "الناتج: ";

    printf("%s\n", str_0);
    scanf("%d", &س);
    t0 = س;
    t1 = 2;
    t2 = t0 * t1;
    t3 = 5;
    t4 = t2 + t3;
    ن = t4;
    printf("%s\n", str_1);
    printf("%d\n", ن);
    return 0;
}
```

The code uses Arabic identifiers like "س" and "ن" instead of standard English ones. The file is named tmpB763\_output.c.

يتم تحويل الكود العربي إلى لغة C القياسية، مع تحويل أسماء المتغيرات العربية (مثل "س") إلى أسماء متوافقة في C (مثل "var\_س"). يتم استخدام مترجم C خارجي (مثل GCC أو TCC) لإنتاج الملف التنفيذي النهائي (exe).

▪ - بيئة التطوير العربية example3.arabic \*

غيل الأمثلة تشفير Lexer ترجمة فقط حفظ فتح تنفيذ جديد

```
// example3.arabic
برنامح حلقة؛
متغير العداد : صحيح؛
متغير المجموع : صحيح؛
المجموع = 0;
العداد = 1;

طالما (العداد <= 10) فان
    المجموع = المجموع + العداد;
    العداد = العداد + 1;
نهاية طالما;

اطبع("مجموع الأعداد من 1 إلى 10 هو: ");
اطبع(المجموع);
```

## مثال تطبيقي كامل - من المصدر إلى التنفيذ

■ - example3.arabic \* - بيئة التطوير العربية

تشغيل الأمثلة □ تشخيص □ ترجمة فقط □ ترجمة وتنفيذ □ حفظ كـ فتح □ حفظ □ جديد □ جمع المخرجات: نوع الإخراج

```
// example3.arabic
برنامـج حلقة؛

متغير العـداد : صحيح؛
متغير المـجموع : صحيح؛

المـجموع = 0;
الـعدد = 1;

طالـما (الـعدد <= 10) فـإن
    المـجموع = المـجموع + الـعدد؛
    الـعدد = الـعدد + 1;
نـهاية طالـما؛

اطـبع ("مجموع الأـعداد من 1 إـلى 10 هـو: ");
اطـبع (المـجموع);
```

==== نتائج الترجمة =====

جارـي تحلـيل المـلف C:\Users\USER\AppData\Local\Temp\tmpEA9.tmp طـول النـص: 407 حرـف

الـتحليل النـحوي مـ بنجـاح! (54 رـمز) ◁

[DBG] Parser::parse() starting with 54 tokens

[DBG] program name='حلقة' next token=SEMICOLON (';')

[DBG] parsing declaration, current token=VARIABLE ('متغير') (الـعدد)

[DBG] parseVariableDeclaration: expecting IDENTIFIER, got IDENTIFIER ('الـعدد') at line=1

[DBG] name='الـعدد' next=COLON (':')

[DBG] saw ':' then token=INTEGER ('صحيح') (نـوع: صحيح)

[DBG] parsed structured type: نوع: صحيح (متغير)

[DBG] parsing declaration, current token=VARIABLE ('المـجموع') (المـجموع)

[DBG] parseVariableDeclaration: expecting IDENTIFIER, got IDENTIFIER ('المـجموع') at line=2

[DBG] name='المـجموع' next=COLON (':')

[DBG] saw ':' then token=INTEGER ('صحيح') (نـوع: صحيح)

[DBG] parsed structured type: نوع: صحيح (المـجموع)

[DBG] parsing statement, current token=IDENTIFIER ('الـمجموع') at line=7

[DBG] parseStatement at token=IDENTIFIER ('الـمجموع') line=7

[DBG] parsing statement, current token=IDENTIFIER ('الـعدد') at line=8

[DBG] parseStatement at token=IDENTIFIER ('الـعدد') line=8

[DBG] parsing statement, current token=WHILE ('طالـما') at line=10

[DBG] parseStatement at token=WHILE ('طالـما') line=10

[DBG] parseStatement at token=IDENTIFIER ('الـمجموع') line=11

[DBG] parseStatement at token=IDENTIFIER ('الـعدد') line=12

[DBG] parsing statement, current token=PRINT ('اطـبع') at line=15

[DBG] parseStatement at token=PRINT ('اطـبع') line=15

[DBG] parsing statement, current token=PRINT ('اطـبع') at line=16

[DBG] parseStatement at token=PRINT ('اطـبع') line=16

الـتحليل النـحوي مـ بنجـاح!

جارـي توـلـيد المـخرجـات ◁

تم توـلـيد الكـود الوـسيـط في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output\_intermediate

تم توـلـيد الكـود الوـسيـط في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output\_intermediate.t

في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.c تم توـلـيد كـود

في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.c تم توـلـيد كـود

تم توـلـيد كـود الجـمـيع في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.asm

تم توـلـيد كـود الجـمـيع في C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.asm

الـكـود الوـسيـط المـولـد ◁

الترجمـة تـمت بنـجـاح

الـسـطر: 9، العمـود: 1

## مثال تطبيقي كامل - من المصدر إلى التنفيذ

■ - بيئة التطوير العربية example3.arabic \*

جميع المخرجات نوع الإخراج تشغيل الأمثلة Lexer تشخيص ترجمة فقط ترجمة وتنفيذ حفظ فتح جديد

```
// example3.arabic
برنامح حلقة؛

متغير العداد : صحيح؛
متغير المجموع : صحيح؛

المجموع = 0;
العداد = 1;

طالما (العداد <= 10) فان
    المجموع = المجموع + العداد;
    العداد = العداد + 1;
نهاية طالما؛

اطبع("مجموع الأعداد من 1 إلى 10 هو: ");
اطبع(المجموع);
```

(الكلام الوسيط المرسل):

```
=====
0: LOAD t0, 0
1: STORE المجموع,t0
2: LOAD t1, 1
3: STORE العداد,t1
4: L0:
5: LOAD t2,
6: LOAD t3, 10
7: LOAD t4, 0
8: CMP t2, t3
9: JLE L2
10: JMP L3
11: L2:
12: LOAD t4, 1
13: L3:
14: JZ t4, L1
15: LOAD t5,
16: LOAD t6,
17: ADD t7, t5, t6
18: STORE المجموع,t7
19: LOAD t8,
20: LOAD t9, 1
21: ADD t10, t8, t9
22: STORE العداد,t10
23: JMP L0
24: L1:
25: PRINT STRING, str_0
26: PRINT VARIABLE, المجموع
27: HALT
```

إحصائيات:

- عدد التعليمات: 28
- عدد المتغيرات المؤلفة: 11
- عدد المآخذ: 4
- عدد السلاسل النصية: 1
- عدد الرموز في جدول الرموز: 2
- الترجمة نجحت بنجاح

الملفات الناتجة:

- C:\Users\USER\AppData\Local\Temp\tmpEA9\_output\_intermediate.txt (الكود الوسيط)
- C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.c (كود C)
- C:\Users\USER\AppData\Local\Temp\tmpEA9\_output.asm (كود Assembly)

الترجمة نجحت بنجاح

السطر: 9، العمود: 1

## مثال تطبيقي كامل - من المصدر إلى التنفيذ

في Temp	نوع:	النوع:	نوع:
tmpEA9_output.asm	Assembler Source	C Source File	C Source File
C:\USER\AppData\Local\Temp		C:\USER\AppData\Local\Temp	C:\USER\AppData\Local\Temp
tmpEA9_output.c			
tmpEA9_output_intermediate.txt			
29/10/2025 4:19 PM			
<b>tmpEA9_output_intermediate.txt</b> المفكرة - tmpEA9_output_intermediate.txt عرض تعليمات تنسيق تحرير ملف		<b>tmpEA9_output.c</b> المفكرة - tmpEA9_output.c عرض تعليمات تنسيق تحرير ملف	
<pre>         =====         LOAD t0, 0 :0         t0 المجموع STORE :1         LOAD t1, 1 :2         t1 العداد STORE :3         :L0 :4         العداد,LOAD t2 :5         LOAD t3, 10 :6         LOAD t4, 0 :7         CMP t2, t3 :8         JLE L2 :9         JMP L3 :10         :L2 :11         LOAD t4, 1 :12         :L3 :13         JZ t4, L1 :14         المجموع,LOAD t5 :15         العداد,LOAD t6 :16         ADD t7, t5, t6 :17         t7 المجموع STORE :18         العداد,LOAD t8 :19         LOAD t9, 1 :20         ADD t10, t8, t9 :21         t10 العداد STORE :22         JMP L0 :23         :L1 :24         PRINT STRING, str_0 :25         المجموع,PRINT VARIABLE :26         HALT :27       </pre> <p>جدول الرموز:</p>		<pre>         =====         #include &lt;stdio.h&gt;         #include &lt;stdlib.h&gt;          } ()int main         ;0 = العداد int         ;0 = المجموع int         ;int t0 = 0         ;int t1 = 0         ;int t2 = 0         ;int t3 = 0         ;int t4 = 0         ;int t5 = 0         ;int t6 = 0         ;int t7 = 0         ;int t8 = 0         ;int t9 = 0         ;int t10 = 0         ;char* str_0 = "مجموع الأعداد من 1 إلى 10 هو:"          ;t0 = 0         ;t0 = المجموع         ;t1 = 1         ;t1 = العداد         :L0         ;t2 = t0         ;t3 = 10         ;t4 = 0       </pre>	
		<b>tmpEA9_output_intermediate.txt</b> تارikh التعديل: 29/10/2025 4:19 PM	

# التقنيات والأدوات المستخدمة

**C++17** </> المترجم الأساسي (ArabicCompiler.exe) - معالجة الكود العربي وتحليله وتوليد الكود الوسيط والنهائي.

**C#/.NET 6.0 WinForms**  بيئة التطوير المتكاملة (ArabicIDE.exe) - واجهة المستخدم الرسومية العربية.

**CMake**  إدارة عملية بناء المترجم وتوليد ملفات المشروع للمنصات المختلفة.

**GCC/TCC**  مترجم خارجي لتحويل كود C المؤلف إلى ملف تنفيذي نهائي.

**أنظمة التشغيل**: Windows/Windows/Linux/macOS لـ IDE، مع دعم بناء المترجم على .



# هيكل المشروع (ملفات ومجذبات)

أخرى	/IDE	Compiler/src
CMakeLists.txt	MainForm.cs	Lexer.h/.cpp
/output (الملفات الناتجة)	Program.cs	Parser.h/.cpp
( اختياري ) /build	ArabicIDE.csproj	AST.h
		Compiler.h/.cpp
		main.cpp

**ملاحظة:** يعتمد المشروع على بنية منظمة تفصل بوضوح بين مكونات المترجم (C++) وواجهة المستخدم (#C) مع الحفاظ على بساطة التصميم وقابلية الصيانة.

# الكلمات المحوزة والميزات اللغوية

الكلمات المحوزة الأساسية:

الكلمة	الوظيفة	الكلمة	الوظيفة
اطبع	أمر الطباعة للإخراج	بداية تعریف البرنامج	برنامیج
اقرأ	أمر القراءة من الإدخال	تعريف متغير جديد	متغير
طالما	حلقة التكرار while	تعريف ثابت	ثابت
كرر	بداية حلقة repeat	بداية الجملة الشرطية	اذا
حتى	شرط إنتهاء حلقة repeat	الجزء التنفيذي عند تحقق الشرط	فإن
نهاية	نهاية الكتلة البرمجية	الجزء التنفيذي عند عدم تتحقق الشرط	وإلا

الخصائص اللغوية:

اسماء عربية للمتغيرات مع تحويل آمن للأسماء في كود C.



تعييرات حسابية ومنطقية قياسية متوافقة مع المعايير العالمية.



كتل وتعليمات تحكم (if/else، loops) بصيغة عربية سهلة الاستخدام.



تحويل مباشر إلى لغة C مع الحفاظ على بنية الكود وأداءه.



# أنواع المخرجات المدعومة

**Assembly**  (اختياري/توضيحي): تمثل منخفض المستوى للبرنامج باستخدام تعليمات شبيهة بلغة التجميع MIPS، يساعد في فهم عملية الترجمة.

**C** (الكود النهائي للتنفيذ): يتم توليد كود C كمرحلة نهائية، ويمكن تنفيذه باستخدام مترجم C مثل TCC أو GCC أو TCC، مما يوفر مرونة وكفاءة في التنفيذ. </>

**الكود الوسيط IR:** تمثل مستقل عن اللغة يتوسط بين الكود المصدري والكود النهائي، يساعد في تطبيق التحسينات وتحليل البرنامج. 

## ملحوظات:

إمكانية عرض كل نوع في نافذة مخرجات IDE مع تلوين للأخطاء والتحذيرات. 

دعم تنفيذ مباشر من الواجهة باستخدام F5. 

# النتائج والإنجازات

-  **بناء مترجم أولي:** يعمل بكفاءة مع قواعد اللغة العربية المحددة في المشروع.
-  **تطوير IDE عربي:** بيئة تطوير متكاملة كاملة الوظائف لعرض جميع مراحل الترجمة بواجهة سهلة الاستخدام.
-  **نجاح التوليد والتنفيذ:** توليد كود C بشكل صحيح والتنفيذ الناجح عبر مترجمات GCC/TCC.
-  **أمثلة عملية:** تطوير نماذج توضيحية تبين الدورة الكاملة من الكود المصدري العربي إلى المخرجات النهائية.
-  **قابلية التوسيع:** تصميم مرن يسمح بإضافة تحسينات وتحليلات مستقبلية لتطوير اللغة.

# الخلاصة والشكر

 **الخلاصة:** المشروع يتيح إمكانية إنشاء منظومة تطوير عربية شاملة (IDE + Compiler) مع الترجمة إلى C، مما يسهل برمجة التطبيقات باستخدام اللغة العربية.

 **أعمال مستقبلية:** تحسين المُحلّلات، إضافة تحسينات (Optimizations)، دعم المزيد من تراكيب اللغة وإدارة المكتبات والحزام، وتطوير واجهة مستخدم أكثر تكاملاً مع أدوات التطوير الأخرى.

 **الشكر والتقدير:** نقدم بجزيل الشكر للدكتور / خالد الكحصه على توجيهاته القيمة وإشرافه على المشروع، ولإدارة جامعة إب وكلية العلوم التطبيقية وقسم علوم الحاسوب وتقنية المعلومات على دعمهم المستمر، ولكل من ساهم في إنجاح هذا المشروع.