بسم الله الرحمن الرحيم

جامعة النيل الأزرق

كلية الدراسات الإضافية

أساسيات البرمجة

أساسيات لغة البيسك

الجزء الثاني

إعداد : أ. عبد الرحمن عباس إبراهيم

2007

بسم الله الرحمن الرحيم

مقدمة:

في الجزء الأول تناولنا خطوات حل المسائل البرمجية ، و كان التركيز أكثر على الثلاث خطوات الأولى باعتبارها الشق اليدوي من حل المسالة بينما الثلاثة المتبقية فتمثل الشق الآلي في حل المسائل – باستخدام الحاسب. تمثل لغة البيسك الأداة التي سنستخدمها في عملية كتابة و ترجمة و اختبار و تنفيذ البرامج – الخطوات الثلاثة الأخيرة– لذا لابد من معرفة أساسيات لغة البيسك .

لغة البيسك :

لغة البيسك احدي لغات المستوى الأعلى HLL و هي لغة بسيطة وسهلة يمكن استخدامها لبناء تطبيقات بسيطة ، و كلمة BASIC اختصار للعبارة الإنجليزية Beginner's All-Purposes Symbolic Instruction Code. أي شفرة تعليمات رمزية لجميع أغراض المبتدئين، و هذا يعنى أنها لغة برمجة سهلة الاستخدام للمبتدئين و أنها لغة تستخدم للأغراض عامة ، غير مخصصة لمجال معين كما في لغة التطبيقات التجارية COBOL.

لغة البيسك كأي لغة تتكون من عدد الرموز و الحروف و المفردات و الكلمات الدالة ، و كذلك على عدد من العبارات و التعبيرات ذات المعنى المحدد .

الحروف الأبجدية للغة البيسك BASIC Alphabet:

تتألف لغة البيسك من عدد من الرموز و الحروف :

1. الحروف الأبجدية الإنجليزية من A إلى Z(حروف كبيرة) و من a إلى z (حروف صغيرة) .
2. الأرقام من 0 إلى 9 .
3. الرموز و العلامات الخاصة مثل + ، = $%^&!@... الخ.

الكلمات المحجوزة Reserved words :

وهى كلمات ذات معانى برمجية تسمى الكلمات الرئيسية (main words) أو الكلمات المفتاحية (key words) و هي كلمات قياسية سابقة التعريف في لغة البيسك ويمكن استخدامها للغرض المحدد لها فقط ولا يمكن استخدامها كمعرفات يعرفها المبرمج كما لا يمكن استخدامها كمتغيرات وذلك حتى لا يحدث بسببها ارتباك إثناء عملية الترجمة و من هذه الكلمات:

if , else ,end if , case , for, next, goto , , do , end

المعرفات Identifier :

هي تلك الأسماء التي تعطى لعناصر البرنامج المختلفة مثل : المتغيرات و الثوابت ,الدوال , المصفوفات .

و للمعرفات شروط لكتابتها:

1. أن يحتوى على حروف أبجدية أو أرقام و يجوز استخدام علامة الشرطة التحتية (_) (under score) .
2. أن يكون أول رمز في المعرف حرفا أبجديا ويسمح باستخدام الحروف الصغيرة أو الكبيرة .
3. يجب أن يكون المعرف خاليا من الرموز الخاصة ماعدا (_)
4. يفضل أن يكون للمعرف الطول المناسب وان يكون واضحا وذا معنى ومدلول

أمثله صحيحة للمعرفات:

REAL , X , area10 , Y15 , tax_rate, Sum_4 , total_value ,

أمثلة على أسماء معرفات غير صحيحة

&address , 5th , last name , order-no , char , "a"

الثوابت constant:

الثوابت معرفات تأخذ قيمة واحدة و تظل ثابتة طوال زمن تنفيذ البرنامج و لا يمكن تغيير قيمتها ، يتم الإعلان عن الكميات الثابتة عن طريق الكلمة المحجوزة const كالتالي :

Const ident=value

حيث ident اسم المعرف الثابت و value القيمة الثابتة

مثلاً const pi = 3.14

المتغيرات Variables:

المتغيرات وعاء لتخزين البيانات الرقمية و الرمزية في الذاكرة أثناء عمل البرنامج و هذا المتغير تستطيع تغيير قيمته في أي لحظة أثناء تنفيذ البرنامج ، لكل متغير نوع بيانات يحدد حجم البيانات بالبايت و مدي قيمة المتغير .

أنواع البيانات Data Types:

تنقسم أنواع البيانات في لغة البيسك إلى:

- البيانات العددية Numerical data type و تضم نوعين :
- الأعداد الصحيحة : و هي الأرقام الخالية من الفاصلة العشرية و تضم نوعين :
  1. الأعداد الصحيحة integer و تأخذ 2 بايت من الذاكرة.
  2. الأعداد الصحيحة الكبيرة Long و تأخذ 4 بايت من الذاكرة.
- الأعداد الحقيقية ذات النقطة العائمة floating point (الفاصلة العشرية) و تضم أيضا نوعان :
  1. الأعداد الحقيقية ذات الدقة المفردة single .
  2. الأعداد الحقيقية ذات الدقة المزدوجة double.
- بيانات رمزية (نصية) string .

بصورة عامة تميز لغة البيسك بين المعرفات العددية و الرمزية بإضافة علامة $ قبل المعرفات الرمزية و حذفها من المعرفات العددية.

مثلاً :

NO=100 متغير رقمي

و $name="Ahmed" متغير رمزي.

نوع البيانات		كلمة التعريف	حجم الذاكرة
عددية صحيحة	صحيح عادي	integer	2byte(16 bit)
	صحيح طويل	long	4 byte(32 bit)
عددية حقيقية	دقة مفردة	single	4 byte(32 bit)
	دقة مزدوجة	double	8 byte(64 bit)
رمزية	-	string	-

كيفية الإعلان عن المتغيرات و المعرفات :

يتم الإعلان عن المعرفات (المتغيرات) باستخدام الكلمة المحجوزة Dim بالصيغة العامة التالية:

Dim var-name AS data_type

حيث :

Var_name: اسم المتغير.

Data type: نوع البيانات.

Dim,as : كلمات محجوزة .

أمثلة :

```
Dim no as integer
Dim phone as long
Dim x as single
Dim y as float
```

يمكن الإعلان عن أكثر من متغير في عبارة واحدة :

```
Dim a,b,c,d as integer
```

الأنواع المعرفة بواسطة المستخدم:

يمكن للمستخدم أن يقوم بتحديد أنواع بيانات جديد باستخدام الكلمة المحجوزة type كالتالي:

```
TYPE Tname
Var1 as data type
Var2 as data type
End Type
```

المؤثرات Operators:

- **المؤثرات الحسابية Arithmetical Operators:**

وهي المؤثرات المسئولة إجراء العمليات الحسابية :

المؤثر	المعني
+	عملية الجمع
-	عملية الطرح
*	عملية الضرب
/	عملية القسمة
mod	باقي القسمة
\	القسمة الصحيحة
^	الأس

أولوية العمليات الحسابية :

عند إجراء عملية حسابية معقدة (تحتوي على عدة مؤثرات حسابية) فان المترجم يقوم بالتنفيذ حسب أولوية العمليات كما بالجدول التالي:

المؤثر	المعني
()	الأقواس
^	الأس
*	عملية الضرب
/	عملية القسمة
mod	باقي القسمة
\	القسمة الصحيحة
+	عملية الجمع
-	عملية الطرح
=	التخصيص

المؤثر mod يعيد باقي القسمة مثلاً 5 mod 3=2 أما المؤثر \ فيعيد العدد الصحيح في عملية القيمة مثلاً 5/3=1

- **المؤثرات المنطقية Logical Operators:**

المؤثرات المنطقية تمكننا من مقارنة تعبيرين و هي:

المؤثر	المعني
and	المؤثر and
or	المؤثر OR
not	المؤثر not

- **المؤثرات العلائقية Relational Operators:**

تمكنا المؤثرات العلائقية من مقارنة قيم رقمية أو حرفية و يكون دائماً ناتج التعبير المنطقي أو العلائقي صحيح أو خطأ True – false(0 أو 1) .

المؤثر	المعني
<	أقل من
>	أكبر من
=<	أقل من أو يساوي
>=	أكبر من أو يساوي
<>	لا يساوي
=	يساوي

التعابير Expressions:

تمثل التعابير الأجزاء الرئيسية التي تتكون منها الجمل البرمجية للغة فكل تعبير يحتوي على مجموعة من المؤثرات يمثل تعبير و هنالك نوعان من التعبيرات :

- **التعابير الحسابية Arithmetical Expression:**

و هي التعابير التي تحتوي على مؤثرات حسابية مثلاً x+y*(z-3)+x/2 و دائماً ناتج التعبير الحسابي قيم رقمية.

- **التعابير المنطقية Logical Expression :**

و هي التعابير التي تحتوي على مؤثرات منطقية مثلاً x<=10

(x>5)&&(y==3), و دائماً ما تكون نواتج هذه التعابير صحيح أو خطأ و تمثل في لغة البيسك بـــ(1,0) .

العبارات statements :

تمثل العبارات التعليمات المكونة للبرنامج المصدر(source program) الذي يقوم المترجم بتحويله إلى البرنامج الهدف (object program) الذي يمثل الحل للمسألة البرمجية التي من اجلها كتب البرنامج.

هنالك العديد من الجمل البرمجية في لغة البيسك منها ما هو مطلوب لإدخال بيانات و أو طباعة مخرجات أو إضافة تعليقات أو التحكم في مسار تنفيذ البرنامج فيما يلي عدد من هذه العبارات :

- **التعليقات Comments:**

التعليقات تستخدم لكتابة عبارات توثيقية توضح تسلسل البرنامج و تزيل اللبس عن الجملة البرمجية و تسهل فيم البرنامج ككل. يتجاهل المترجم التعليقات و لا يقوم بترجمتها ، لكتابة تعليق نستخدم الكلمة المحجوزة dim و نكتب التعليق بهدها أو نستخدم الرمز ’ (الفاصلة العليا).

مثلاً:

Dim here you can add your comment

أو

’here you can add your comment

- **عبارات الإدخال و الإخراج** input & out put statements :

أولاً عبارات الإدخال :

و هي العبارات المسئولة عن إدخال القيم الرقمية و الرمزية إلى البرنامج ، و تضم لغة البيسك العبارات :

1- عبارة read-data:

تستخدم هذه العبارة في حال أن البيانات متوفرة داخل البرنامج و لا يتم إدخالها من وحدة الإدخال النمطية (key board) أثناء تنفيذ البرنامج ، الصيغة العامة لهذه العبارة:

Read list of var
…..
…..
…..
Data list of values

List of var: قائمة المتغيرات،

List of values : قائمة القيم.

Read, data: كلمات مفتاحية.

ليس شرطاً أن تتقدم عبارة read علي data و لكن شرطاً أن تكون قائمة المتغيرات مساوية لقائمة القيم .

مثال:

```
Read x,y,z
…..
Data 10,5,3
```

2- عبارة Input:

تمنك هذه العبارة من إدخال بيانات أثناء تنفيذ البرنامج من وحدة إدخال نمطية مثل لوحة المفاتيح ، كما تتيح التفاعل بين المستخدم و البرنامج ، حيث يقوم البرنامج بمخاطبة المستخدم عبر رسالة ،و يقوم المستخدم بإدخال البيانات المطلوبة . الشكل العام لهذه العبارة:

```
Input"message",list of var,
```

تمثل message رسالة التفاعل بين البرنامج و المستخدم وهي اختيارية ،يمكن حذفها. و تمثل list of var قائمة المتغيرات ، قد يكون متغير واحد أو عدة متغيرات .

مثال:

```
Input "enter three numbers",x,y,z
```

ثانياً عبارات الإخراج Output Statements:

عبارات للإخراج مسئولة عن عرض المعلومات على وحدات الإخراج مثل الشاشة و الطابعة ، من عبارات الإخراج:

1- عبارة PRINT :

تستخدم هذه العبارة لطباعة المخرجات على الشاشة ، يمكنها أن تتطبع قيم رقمية أو سلسلية ، الصيغة العامة :

```
PRINT "MESSAGE ",VAR OR EXP
```

VAR متغير ، EXP: تعبيرMESSAGE رسالة نصية .

أمثلة :

```
PRINT "THE OUTPUT"
PRINT "X=",X;
```

```
PRINT"SUM=",A+B+C
PRINT X,Y,Z
PRINT A*B+C/3
```

من الأمثلة السابقة يتضح أن عبارة PRINT يمكنها طباعة نصوص بشرط وضعها بين علامتي تنصيص "" أو قيم متغيرات أو ناتج تعبيرات حسابية أو منطقية ، أو خليط من كل ما سبق بأي ترتيب.

2- عبارة Lprint:

تستخدم هذه العبارة لطباعة المخرجات على الطابعة بدلاً عن شاشة العرض و تستخدم تماماً كما في عبارة Print.

- **عبارات التخصيص** assignment statements.

تقوم عبارات التخصيص بإسناد و تخصيص قيم رقمية أو رمزية أو منطقية لمتغير ، و يستخدم معها مؤثر التخصيص (=).

الصيغة العامة :var1=value حيث يمثل var1 المتغير و value القيمة المسندة إلى المتغير، في بعض الأحيان تستخدم الكلمة LET في عملية الإسناد مثلاً LET VAR=VALUE .

أمثلة :

```
Radius=15;
Let x=105
```

تستخدم عبارة الإسناد أيضاً في حساب ناتج تعبير حسابي أو منطقي

مثلاً SUM=A+B+C أو LET SUM=A+B+C

أمثلة محلولة (2-1):

أكتب برنامج بلغة البيسك ل:

1. طباعة ناتج التعابير التالية :

```
1- A+B*C/2-5*B
2- (A<B)AND(C<5)OR(B=10)
3- A^2+B*C
```

اذاكان A=3,6=10,C=6

2. لطباعة العبارة (الرسالة)التالية:

WELCOME TO BASIC PROGRAMMING

3. .لقراءة 3اعداد وطباعة مجموعها
4. لقراءة 5 إعداد وحساب متوسطها
5. لحساب مساحة الدائرة =AREA

الحلول:

البرنامج الأول:

```
Cls
A=3
B=10
C=6
Exp1=A+B*C/2-5*B
Exp2=(A<B) and(c<5)Or(B=0)
Exp3=A^2+2*B*C
Print “Exp1 =”,Exp1
Print “Exp2 =”,Exp2
Print “Exp3 =”,Exp3
End.
```

المخرجات :

```
Exp1 = -17
Exp2 = 0
Exp3 = 129
```

البرنامج الثاني:.

```
Print “Welcome To Basic Programming"
End.
```

المخرجات

```
Welcome To Basic Programming
```

البرنامج الثالث :

```
Input “Enter Three Numbers”,A,B,C
```

```
Sum=A+B+C
Print "the summation = ",sum
End
```

المخرجات

```
Enter Three Numbers 1,2,3
the summation = 6
```

البرنامج الرابع:

```
Cls
Input "please enter 5 numbers" , A,B,C,D,E
Sum=A+B+C+D+E
AV=sum/5
Print "the Average =",AV
End
```

المخرجات

```
Please Enter 5 Numbers 1,2,3,4,5
the Average = 3
```

البرنامج الخامس:

```
const pi=3.14
input "enter the Radius " , R
area = pi*R^2
print "the area =" , area
end
```

إعادة كتابة البرنامج الأول باستخدام عبارة read .. data

```
Cls
Read A,B,C
Exp1=A+B*C12-5*B
Exp2=(A<B) and(c<5)Or(B=0)
```

```
Exp3=A^2+2*B*C
Print "Exp1 =",Exp1
Print "Exp2 =",Exp2
Print "Exp3 =",Exp3
Data 3,10,6
End.
```

- **عبارات التحكم** Control Statements:

عند تنفيذ أي برنامج يبدأ التنفيذ بصورة متسلسلة من الأعلى إلى الأسفل ، للتحكم في مسار تنفيذ البرنامج سواء بتنفيذ تعليمات تحت شروط محددة أو تكرار تنفيذ تعليمات معينة ، يمكن تقسيم عبارات التحكم في لغة البيسك إلى قسمين رئيسيين :

- عبارات الشرط (التفرع) branching statement.
- عبارات التكرار loop statements.

أولاً : عبارات التفرع:

- **عبارة IF:**

هي أشهر عبارات التحكم، تستخدم للاختيار و تنفيذ عبارات وفقاً لشرط محدد، الصيغة العامة للعبارة :

```
If Condition THEN
STATEMENT
Else
Statements
END IF
```

condition: تعبير منطقي يحتمل الصواب أو خطأ.

Statements: عبارة أو عبارات سيتم تنفيذها.

Else: تنفذ التعليمات التالية لها إذا كان ناتج التعبير المنطقي خطأ.

عند تنفيذ عبارة if..then فان المترجم ينظر للتعبير الشرطي condition فإذا كان ناتجه صواب (1) فان العبارات التالية لها ستنفذ مباشرة ، أما إذا كان ناتج التعبير المنطقي خطأ(0) فإن العبارات التالية لعبارة else سيتم تنفيذها . يمكن الاستغناء عن عبارة else اذا كانت هنالك عبارات محددة سيتم تنفيذها إذا تحقق الشرط ، و لا توجد عبارات أخرى في حال عدم تحقق الشرط(التعبير المنطقي) .

مثلاً

```
x=5
if x<7 then
print “less than seven”
else
print” greater than or equal 7”
end if
```

الشفرة السابقة ستطبع العبارة الأول "less than seven " لأن ناتج التعبير المنطقي (صحيح 1). أما إذا استبدلنا التعبير المنطقي بـــ(x>7) فان عبارة printالثانية سيتم تنفيذها لأن ناتج التعبير سيصبح (خطأ 0).

- **عبارة select case:**

هذه العبارة توفر للمستخدم عدة خيارات لتنفيذها ، العبارة if توفر خيارين فقط أما select فتوفر خيارات عديدة تحت شروط عديدة:

الصيغة العامة

```
Select case(var)
Case val1:statement1
Case val2:statement2
Case val3:statement3
Case else:stat_def;
Select end
```

حيث تمثل (select case) عبارة تمرير الخيار أو الشرط الذي سيختاره المستخدم و المتغير (var) القيمة التي ستمرر عبر العبارة select case بينما عبارات case المتكررة بقيم مختلفة case val1,2,3 فستحدد العبارة التي سيتم تنفيذها وفقاً للقيمة الممررة عبر var. أما stst_def فهي العبارة الافتراضية التي يستم تشغيلها في حال عدم تتطابق القيمة الممرة مع أيٍ من قيم case .

مثلاً:

```
Select case (5)
Case 1: print” one”
Case 2:print “two”
Case 5:print”five”
Case else :print”none”
Select end
```

العبارات السابقة ستطبع "five" للتطابق القيمة الممررة 5 مع قيمة عبارة case 5 .

ثانياً : عبارات التكرار loop statement:

في بعض البرامج قد نحتاج إلى تكرار عبارة برمجية ما عدة مرات ، مثلا لطباعة عبارة "welcome" مائة فإننا سنحتاج إلى مائة عبارة إخراج ، و هذا ليس سيأخذ وقت طويل في الكتابة و سيحجز مساحة أكبر في الذاكرة .

عبارات التكرار توفر لنا الوقت و المساحة، وتساعدنا على تكرار عبارة أو عبارات عدة مرات باستخدام جمل برمجية صغيرة . من هذه العبارات :

عبارة for .. next

عبارة for احدي عبارات التكرار تساعد على تكرار تنفيذ عدة عبارات ، الصيغة العامة لهذه العبارة :

```
For count= init_val to max_val step val
Ststemants
Next count
```

حيث :

For,to ,step,next: كلمات محجوزة .

Count : متغير رقمي مرتب (integer ,long) يحسب عدد الدورات و يسمى العداد.

Init_val القيمة الأولية للعداد count.

Max_val: القيمة النهائية التي سيتوقف عندها الدوران.

Step: مقدار الزيادة في العداد ،إذا لم تكتب فان المترجم سيعتبرها 1.

Statement: العبارات التي سيتم تكرارها.

مثلاً لطباعة عبارة welcome مائة مرة :

```
For i= 1 to 100
Print “welcome”
Next i
```

لاحظ المتغير العداد I و معدل الزيادة 1 لعدم كتابة الخطوة step ، و العبارة next تنهي الدورة الحالي و تنشئ دورة جديدة إذا لم تصبح قيمة العداد مساوية للقيمة العظمى max_val.

- **عبارة while wend**

عبارة while أيضا من عبارات التكرار ، و لكنها تقم بتكرار عمليات وفقاً لشروط محددة ، الصيغة العامة لهذه العبارة:

```
while condition
statement
wend
```

حيث:

While wend: كلمات محجوزة ، تحدد الأولى شرط التكرار ، و تحدد الثانية نهاية العبارات التي سيتم تنفذها.

Statements العبارات التي سيتم تنفيذها

مثلاً لطباعة عبارة welcome 100 مرة باستخدام while :

```
i=0
While i<100
Print "welcome"
i=i+1
wend
```

لاحظ العداد I يجب وضع قيمة ابتدائية له (i=0) و تحديد مقدار الزيادة له (i=i+1) .

- **عبارة do while**

تشابه كثير عبارة while wend و تختلف عنها في الصيغة العامة:

```
Do while condition
Statements
loop
```

فقط باستبدال wend ب loop و إضافة do، نفس المثال السابق باستخدام do while:

```
i=0
do While i<100
Print "welcome"
i=i+1
loop
```

- **عبارة do until:**

تختلف عبارة do until عن do while و while wend في أنه إذا كانت قيمة التعبير المنطقي true فإن do until ستتوقف عن التكرار بينما تستمر في العبارات الأخرى.

الصيغة العامة :

```
Do until condition
Statements
loop
```

do until condition : تحدد شرط التوقف (أفعل حتى يتحقق الشرط condition) ، و loop استمر.

المثال السابق باستخدام do until:

```
i=0
do until i=100
Print "welcome"
i=i+1
loop
```

لاحظ الشرط تغير إلى i=100 أي توقف عندما تصبح i=100.

أمثلة محلولة (2-2)

1- اكتب برنامج لقراءة رقمين و طباعة الرقم الكبر.

2- أكتب برنامج لاختيار قيمة رقمية بين1 – 6 وطباعة الرقم باللغة الإنجليزية باستخدام select case .

3- أكتب برامج بلغة البيسك لطباعة :

1-الأعداد من 1 إلى 20

2-الأعداد الفردية بين 1-20

باستخدام :

1- for:

2- While wend

3- Do while

4- Do until

الحلول:

البرنامج الأول

```
Cls
Input”enter two numbers”,x,y
If x>y then
Print X ,“is greater then”, Y
Else
Print X ,“is greater then”, Y
End if
End
```

المخرجات

```
Enter two numbers 5,8
8 is greater  than 5
```

البرنامج الثاني:

```
Cls
Input” enter your choice”,ch
Select case(ch)
Case 1:print” you choose one”
Case 2:print” you choose two”
Case 3:print” you choose three”
Case 4:print” you choose four”
Case 5:print” you choose five”
Case 6:print” you choose six”
Case else :print “out of choices”
Select end
```

المخرجات

```
enter your choice 4
you choose four
```

البرامج:

أ. طباعة الأعداد من 1 -100

1- باستخدام for

```
cls
for i= 1 to 20
print i
next i
```

```
end
```

2- **باستخدام** while wend

```
count=1
while count<=20
print count
count=count+1
wend
end
```

3- **باستخدام** do while

```
cls
count=1
do while count<=20
print count
count=count+1
loop
end
```

4- **باستخدام** do until

```
count=1
do until count>20
print count
count=count+1
loop
end
```

المخرجات

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

ب. طباعة الأعداد الفردية بين 1 -100

5- باستخدام for

```
cls
for i= 1 to 20 step 2
print i
next i
end
```

6- باستخدام while wend

```
cls
count=1
while count<=20
print count
count=count+2
wend
end
```

7- باستخدام do while

```
cls
count=1
do while count<=20
print count
count=count+2
loop
end
```

8- باستخدام do until

```
cls
count=1
do until count>20
print count
count=count+2
loop
end
```

المخرجات

```
1
3
5
7
9
11
13
15
```

الدوال functions

تذخر مكتبة لغة البيسك بالعديد من الدوال (مجموعة برامج فرعية جاهزة يستخدمها المبرمج لحساب قيم مثل الجزر التربيعي)، من دوال مكتبة لغة البيسك :

اسم الدالة	الوظيفة	مثال
Abs()	حساب القيمة المطلقة لرقم	Abs(-5) = 5
Sqr()	حساب الجزر التربيعي	Sqr(9) = 3
Sin()	حساب جيب الزاوية بالراديان	Sin(3.14)= 0
Log()	حساب اللوغريتم لعدد	Log(1) = 0
Len()	حساب طول سلسلة	Len("ali") = 3

كل دالة من الدوال السابقة تأخذ قيمة (وسائط – دائما تكون بين الأقواس) و ترجع قيما .

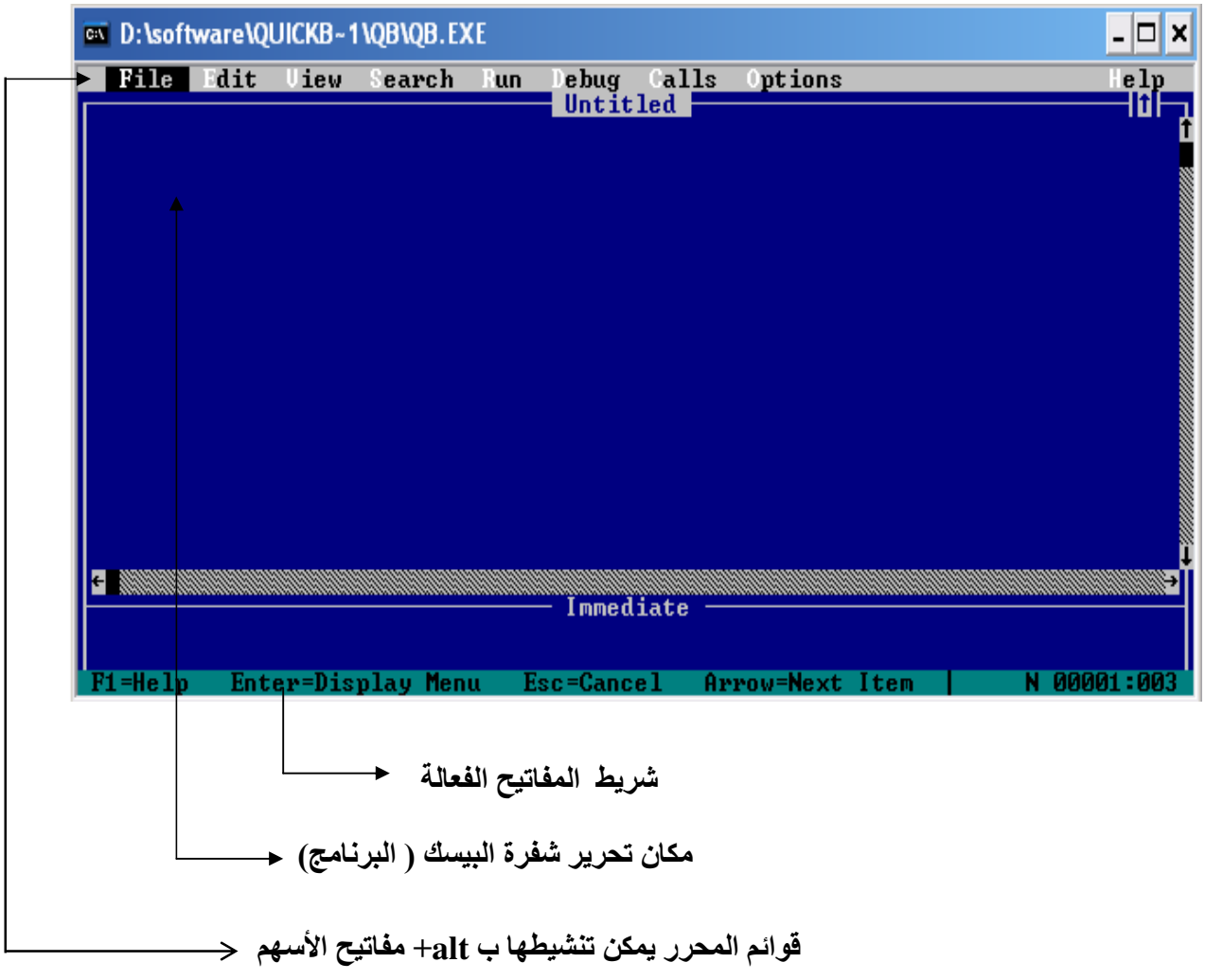
محررات لغة البيسك

هنالك العديد من إصدارات لغة البيسك ، كل إصدار له محرر يختلف عن الإصدار الآخر من هذه الإصدارات :

- BASIC
- Quick basic
- GW basic
- Basica

محرر Quick basic:

يتمتع محرر البيسك السريع بالعديد من المزايا تفوق بها على المحررات الأخرى ،باحتوائه على قوائم ترتب كل العمليات ، مثل عمليات إنشاء و حفظ الملفات ، عمليات التحرير ، و البحث و الاستبدال وعمليات التصحيح و التدقيق و التنفيذ ، الشكل التالي يوضح محرر البيسك السريع quick basic:



الجدول التالي يبين كيفية استخدام مفاتيح لوحة المفاتيح في عمليات التحرير و التنقيح و ترجمة و تنفيذ البرامج:

الهدف	الإجراء
إنشاء ملف جديد	القائمة file ،ثم الأمر new
حفظ ملف	القائمة file ،ثم الأمر save

فتح ملف موجود	القائمة file ،ثم الأمر open
كاتبة شفرة برمجية	استخدم مفاتيح الأحرف و الأرقام و العلامات الخاصة
تشغيل البرنامج	القائمة run الأمر start أو المفاتيح shift+F5
طلب المساعدة	القائمة help أو الضغط علي المفتاح F1

وظائف مفاتيح لوحة المفاتيح :

أثناء عمليات التحرير قد نحتاج لبعض أو كل المفاتيح التالية :

المفتاح	الوظيفة
alt	تنشيط القوائم
shift + أي حرف	تبديل حالة الأحرف (small – capital)
home	نقل المؤشر أول السطر
End	نقل المؤشر آخر السطر
سهم أعلى	نقل المؤشر سطر لأعلى
سهم أسفل	نقل المؤشر سطر أسفل
سهم يمين	نقل المؤشر حرف لليمين
سهم أعلى	سطر لأعلى
delete	مسح رمز لجهة اليمين
Back space أو ←	مسح رمز لجهة الشمال
esc	الهروب (إغلاق مربعات الحوار)
Enter	سطر جديد
Shift+f5	تشغيل البرنامج

انتهى الجزء الثاني بحمد الله

أمنياتي لكم بدوام التفوق و النجاح