

本章通过以下几个案例介绍 VBA 应用技术：国标汉字的输入和代码获取，文本查找，表格操作，图文框应用，状态栏控制，网页信息处理等。

5.1 快速输入国标汉字

我国于 1981 年颁布了信息交换用汉字编码基本字符集的国家标准，即 GB2312，对 6763 个汉字、628 个图形字符进行了统一编码，为信息处理和交换奠定了基础。虽然 Office 2016 中文版支持超大字符集，但在多数情况下，人们用计算机处理的汉字一般都没有超出基本集这 6763 个汉字。

由于某种特殊应用（如打印字帖、打印区位码表等），需要在 Word 文档中输入 GB2312 的全部汉字。一个个从键盘输入既慢又容易出错，显然不是好办法。编写一个 VBA 程序，可以轻松地解决这个问题。

具体做法如下：

进入 Word，在“开发工具”选项卡“代码”选项组中单击“宏”按钮，在“宏”对话框中输入宏名“输入国标汉字”，指定宏的位置为当前文档，单击“创建”按钮，进入 VB 编辑环境，创建以下程序：

```
Sub 输入国标汉字()  
    For m = 176 To 247  
        For n = 161 To 254  
            nm = "&H" & Hex(m) & Hex(n)  
            Selection.TypeText Text:=Chr(nm)  
        Next  
    Next  
End Sub
```

这是一个双重循环结构程序，外层循环得到汉字内码的高位（范围是 176 到 247 之间的整数），内层循环得到汉字内码的低位（范围是 161 到 254 之间的整数）。循环体中，将内码的高位和低位以十六进制数字形式拼接，即得到一个汉字的完整内码，用 Chr 函数将内码转换为汉字，用 Selection.TypeText 方法输入到当前文档。

运行上述程序，便可在当前文档中得到 GB2312 的全部汉字。

5.2 查汉字区位码

为保证汉字信息输入的准确性,有些场合要使用汉字的区位码。例如,填报中考、高考志愿表时,关键内容的汉字信息需要同时填写对应的区位码。通常,区位码可以查表得到,但是如果手头暂时没有区位码表,怎么查找每个汉字的区位码呢?下面的 VBA 程序可以解决这个问题。

进入 Word,在“开发工具”选项卡“代码”选项组中单击“宏”按钮,在“宏”对话框中输入宏名“查汉字区位码”,指定宏的位置为当前文档,单击“创建”按钮,进入 VB 编辑环境,创建以下程序:

```
Sub 查汉字区位码()  
    nm = Hex(Asc(Selection.Text)) '内码(4位十六进制形式)  
    nm_h = "&H" & Left(nm, 2) '内码(高2位)  
    nm_l = "&H" & Right(nm, 2) '内码(低2位)  
    qm = nm_h - 176 + 16 '得到区码  
    wm = nm_l - 161 + 1 '得到位码  
    wm = IIf(wm < 10, "0" & wm, wm) '2位数表示  
    MsgBox qm & wm '显示区位码  
End Sub
```

上述程序段首先取出选定文本(单个汉字),用 ASC 函数求出汉字的内码,用 Hex 函数将汉字的内码转换为 4 位十六进制字符串型数据。然后用 Left 和 Right 函数分别取出内码的高 2 位和低 2 位(用十六进制字符串表示)。最后将内码的高 2 位和低 2 位分别转换为区码和位码并显示出来。

由于位码可能是 1 位数,也可能是 2 位数,因此为使格式规整,可用 IIf 函数统一转换为 2 位数。IIf 函数可以根据条件的真假,返回不同的值,语法形式为:

IIf(条件表达式,条件为 True 时的返回值,条件为 False 时的返回值)

当变量 wm 的值小于 10 时,函数 IIf(wm < 10, "0" & wm, wm)返回一个在 wm 的值前面加上一个字符“0”而形成的字符串。当变量 wm 的值大于或等于 10 时,则返回 wm 自身的值。

要查询某个汉字的区位码,可先在 Word 中选中这个汉字,然后运行“查汉字区位码”子程序,就可得到该汉字的区位码。例如,“博”字的区位码为 1809,“达”字的区位码为 2079。

5.3 免试生筛选

某高校计算机系要对学生软件开发能力考核。规定:如果“计算机导论”“数据库应用”“C 语言”“数据结构”“VB 程序设计”“多媒体技术”“汇编语言”“计算机网络”这 8 门课程的考试成绩中,有 2 门以上(含 2 门)排在全年级前 10%以内,则“软件开发能力”成绩记为 A 等,不必另行参加考核。

假设有一个 Word 文档，其中有 8 个表格，每个表格的内容为一门课成绩排在全年级前 10% 以内的学生名单和相关信息。图 5-1 给出了其中 2 门课对应的表格结构和内容。

学号	班级	姓名	计算机导论
0012431	4	张伟	98.0
0012212	2	陈强	94.0
0012248	2	张红军	94.0
0012104	1	殷春花	92.0
0012145	1	陈立杰	91.0
0012423	4	朱艳平	91.0
0012229	2	何庆新	90.0
0012245	2	董雪	90.0
0012417	4	毛怀勇	88.0
0012105	1	张彦军	87.0
0012112	1	刘丹	87.0
0012316	3	侯萍	87.0
0012426	4	李鸿亮	87.0
0012333	3	栾丽静	86.0
0012424	4	姜艳艳	86.0
0012211	2	张艳芬	85.0
0012318	3	张坦	85.0
0012342	3	李倩	85.0
0012422	4	关金萍	85.0
0012108	1	陈艳超	84.0
0012236	2	刘洪伟	84.0
0012304	3	江超	84.0

学号	班级	姓名	数据库应用
0012105	1	张彦军	92.0
0012423	4	朱艳平	90.0
0012112	1	刘丹	88.0
0012417	4	毛怀勇	88.0
0012421	4	庄延锋	88.0
0012110	1	张亚荣	86.0
0012114	1	赵鑫章	86.0
0012137	1	张雨	86.0
0012424	4	姜艳艳	86.0
0012438	4	王海	86.0
0012248	2	张红军	85.0
0012303	3	任慧	85.0
0012348	3	秦帅	85.0
0012149	1	刘闯	84.0
0012342	3	李倩	84.0
0012131	1	奚丽娜	83.0
0012205	2	逯博	83.0
0012212	2	陈强	83.0
0012305	3	白银玲	83.0
0012313	3	王婷	83.0
0012107	1	李闯	82.0
0012133	1	鞠向伟	82.0

图 5-1 排在全年级前 10% 的单科成绩信息

为了列出符合免试条件的学生名单，需要把 8 个表格中出现过 2 次以上的学生姓名挑选出来。虽然用“查找”等操作可以完成这一任务，但效率不高。下面介绍一种用 VBA 程序提高效率的方法。

1. 编写程序

打开含有图 5-1 所示的 8 门课成绩信息的 Word 文档，进入 VB 编辑环境，在当前工程中插入一个模块。

在模块中输入语句 `Dim stt As String`，声明一个模块级变量。然后，建立以下 3 个子程序：

```

Sub 查次数()
    stt = Selection.Text           ' 选中的文本
    Selection.Find.Text = stt      ' 作为查找的内容
    Selection.HomeKey Unit:=wdStory ' 到文件头
    Selection.Find.Execute         ' 进行查找
    While Selection.Find.Found()   ' 找到，计数，继续
        k = k + 1
        Selection.Find.Execute
    Wend
    MsgBox "该文本出现 " & k & " 次" ' 显示次数
    Selection.Find.Wrap = wdFindContinue ' 回绕查找（光标回原位）
    Selection.Find.Execute
End Sub
Sub 下一处()
    Selection.Find.Text = stt      ' 上一个操作选中的文本

```

```
Selection.Find.Wrap = wdFindContinue '回绕查找
Selection.Find.Execute
End Sub
Sub 删除()
    Selection.SelectRow                '选中表格当前行
    Selection.Rows.Delete              '删除表格当前行
End Sub
```

“查次数”子程序用来在当前文档中查找并显示选定文本出现的次数，然后光标回到原处。“下一处”子程序用来将光标定位到选定文本下一次出现的位置。“删除”子程序用来删除表格当前行。

2. 在快速访问工具栏中添加按钮

为了便于操作，可在 Word 当前文档的快速访问工具栏中添加 3 个按钮，分别用来执行“查次数”“下一处”“删除”子程序。添加方法如下：

(1) 右击 Word 的快速访问工具栏，在快捷菜单中选择“自定义快速访问工具栏”命令，打开图 5-2 所示的“Word 选项”对话框。在“从下列位置选择命令”下拉列表框中选择“宏”项，在“自定义快速访问工具栏”下拉列表中选择“用于‘免试生筛选.docm’”，将左侧列表框中的 3 个宏添加到右侧列表框。

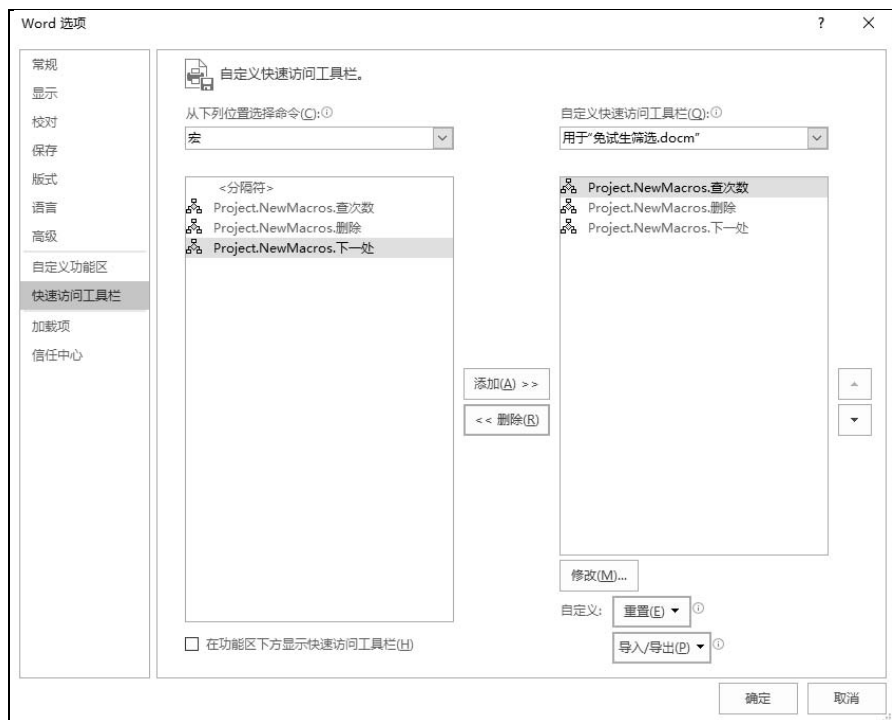


图 5-2 “Word 选项”对话框

(2) 在右侧列表框中选中的一个宏，单击列表框下面的“修改”按钮。在图 5-3 所示的“修改按钮”对话框中，指定按钮的图标符号、修改显示名称，然后单击“确定”按钮。

用同样方法对另外 2 个宏进行修改,最后单击“Word 选项”对话框的“确定”按钮,Word 当前文档的快速访问工具栏中就会出现添加的这 3 个按钮。

3. 名单筛选

在文档中选定一个学号或姓名,单击快速访问工具栏上“查次数”按钮,显示它在文档中出现的次数。如果次数等于 1,就说明该生不符合免试条件,可直接单击“删除”按钮进行删除。如果次数大于 1,则说明该生符合免试条件,应该保留该生一条记录,删除其余的记录。方法是单击“下一处”和“删除”按钮,直至将多余的记录全部删除,再单击“下一处”按钮,将光标定位到原处,以便处理下一条信息。依次处理每条记录,最后保留的就是符合免试条件的学生名单。



图 5-3 “修改按钮”对话框

5.4 名片制作模板

尽管利用 Word 2016 的在线模板可以制作名片,但操作比较复杂,格式也不够灵活。用 Word 2016 的宏创建一个名片制作模板,有一定实用性。本节介绍这个模板的设计和使用方法。

1. 页面及图文框设置

创建一个 Word 文档,保存为启用宏的 Word 文档:“名片制作模板.docm”。

右击 Word 的快速访问工具栏,在快捷菜单中选择“自定义快速访问工具栏”命令,打开“Word 选项”对话框。在“从下列位置选择命令”下拉列表框中选择“不在功能区中的命令”项,在“自定义快速访问工具栏”下拉列表中选择“用于所有文档”,将左侧列表框中的“插入横排图文框”项添加到右侧列表框。单击“确定”按钮,在 Word 左上角的快速访问工具栏上即可看到“插入横排图文框”按钮.

在“布局”选项卡的“页面设置”选项组中,单击对话框启动按钮。在“页面设置”对话框中,选定“纸张”选项卡,指定“宽度”和“高度”分别为 19.5 厘米和 29.5 厘米(标准名片纸张规格)。选定“页边距”选项卡,指定上、下、左、右边距均为 0.7 厘米。单击“确定”按钮,退出“页面设置”。

单击快速访问工具栏的“插入横排图文框”按钮,在当前文档空白处拖动鼠标,添加一个图文框。选中图文框,右击,在快捷菜单中选择“设置图文框格式”命令。在图 5-4 所示的“图文框”对话框中,设置宽度为固定值 8.6 厘米、高度为固定值 5.4 厘米(标准名片规格)。单击“确定”按钮,退出“图文框”设置。

选中图文框,右击,在快捷菜单中选择“边框和底纹”命令。在“边框和底纹”对话框的“边框”选项卡中,设置“无”边框,单击“确定”按钮,将图文框的边框取消。



图 5-4 “图文框”对话框

2. 编写代码

进入 VB 编辑环境，插入一个模块，编写一个“复制及排版”子程序，代码如下：

```
Sub 复制及排版()  
    Selection.Cut                '剪切选中的图文框  
    For k = 1 To 10              '粘贴 10 次  
        Selection.Paste  
    Next  
    Selection.MoveLeft Unit:=wdItem                '选中第 1 个图文框  
    With Selection.Frames(1)  
        .HorizontalPosition = wdFrameLeft          '水平相对于页边距、左侧  
        .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin  
        .VerticalPosition = wdFrameBottom          '垂直相对于页边距、底端  
        .RelativeVerticalPosition = wdRelativeVerticalPositionMargin  
    End With  
    Selection.MoveLeft Unit:=wdItem                '选中第 2 个图文框  
    With Selection.Frames(1)  
        .HorizontalPosition = wdFrameRight          '水平相对于页边距、右侧  
        .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin  
        .VerticalPosition = wdFrameBottom          '垂直相对于页边距、底端  
        .RelativeVerticalPosition = wdRelativeVerticalPositionMargin  
    End With  
End Sub
```

```
Selection.MoveLeft Unit:=wdItem           '选中第 3 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameRight     '水平相对于页边距、右侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = CentimetersToPoints(17.7) '垂直相对于页面、17.7 厘米
    .RelativeVerticalPosition = wdRelativeVerticalPositionPage
End With
Selection.MoveLeft Unit:=wdItem           '选中第 4 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameLeft      '水平相对于页边距、左侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = CentimetersToPoints(17.7) '垂直相对于页面、17.7 厘米
    .RelativeVerticalPosition = wdRelativeVerticalPositionPage
End With
Selection.MoveLeft Unit:=wdItem           '选中第 5 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameLeft      '水平相对于页边距、左侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = wdFrameCenter      '垂直相对于页边距、居中
    .RelativeVerticalPosition = wdRelativeVerticalPositionMargin
End With
Selection.MoveLeft Unit:=wdItem           '选中第 6 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameRight     '水平相对于页边距、右侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = wdFrameCenter      '垂直相对于页边距、居中
    .RelativeVerticalPosition = wdRelativeVerticalPositionMargin
End With
Selection.MoveLeft Unit:=wdItem           '选中第 7 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameRight     '水平相对于页边距、右侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = CentimetersToPoints(6.4) '垂直相对于页面、6.4 厘米
    .RelativeVerticalPosition = wdRelativeVerticalPositionPage
End With
Selection.MoveLeft Unit:=wdItem           '选中第 8 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameLeft      '水平相对于页边距、左侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = CentimetersToPoints(6.4) '垂直相对于页面、6.4 厘米
    .RelativeVerticalPosition = wdRelativeVerticalPositionPage
End With
Selection.MoveLeft Unit:=wdItem           '选中第 9 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameRight     '水平相对于页边距、右侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = wdFrameTop         '垂直相对于页边距、顶端
    .RelativeVerticalPosition = wdRelativeVerticalPositionMargin
```

```
End With
Selection.MoveLeft Unit:=wdItem                                '选中第 10 个图文框
With Selection.Frames(1)
    .HorizontalPosition = wdFrameLeft                          '水平相对于页边距、左侧
    .RelativeHorizontalPosition = wdRelativeHorizontalPositionMargin
    .VerticalPosition = wdFrameTop                              '垂直相对于页边距、顶端
    .RelativeVerticalPosition = wdRelativeVerticalPositionMargin
End With
End Sub
```

在上述子程序中，首先把选中的图文框剪切下来，再粘贴 10 次，得到 10 个相同的图文框。然后，依次选中每个图文框，设置不同的水平和垂直位置，使这 10 个图文框均匀排列到整个页面。

3. 使用方法

用以上方法创建的“名片制作模板.docm”，可以作为一个应用软件，需要时随时打开，用它来设计、打印名片。具体使用方法如下：

- (1) 打开“名片制作模板.docm”。
- (2) 在图文框中输入名片的具体内容，设置字体字号，进行排版，可以插入图片和进行艺术加工，设计出一张名片样板。
- (3) 选中名片样板图文框。
- (4) 运行“复制及排版”子程序，Word 便自动将设计好的名片样板复制 10 份并均匀地排列在整个页面上。
- (5) 将整个页面内容打印输出到名片纸上。
- (6) 用裁纸机将每页纸上的 10 张名片裁剪下来。

5.5 在状态栏中显示进度条

利用 Word 状态栏，可以制作动态的进度条。将这一技术应用到软件当中，能够直观地显示工作进度，改善用户长时间等待的心理状态。

创建一个 Word 文档，保存为“在状态栏中显示进度条.docm”。

进入 VB 编辑环境，在当前工程中插入一个模块，在模块中编写一个“显示进度”子程序，代码如下：

```
Sub 显示进度()  
    wtm = "当前进度: "  
    kk = ""  
    sk = ""  
    ck = Len(kk)           '进度条长度  
    n = 500                '循环次数  
    m = 1 + n \ ck         '每循环 m 次，刷新进度条 1 次  
    For k = 1 To n          '循环  
        Selection.TypeText Text:=Rnd      '模拟要执行的操作  
        Selection.TypeParagraph
```

```

If k Mod m = 0 Then                                'k 为 m 的整数倍
    c = k \ m                                       '进度格数量
    p = Left(sk, c) & Right(kk, ck - c)           '调整进度格
    Application.StatusBar = wtm & p               '更改系统状态栏的显示
End If
DoEvents                                           '转让控制权给操作系统
Next
Application.StatusBar = False                     '恢复系统状态栏
End Sub

```

上述子程序首先用变量 `wtm` 保存字符串“当前进度:”。定义两个变量 `kk` 和 `sk`, 分别保存由空心菱形块和实心菱形块组成的字符串, 并求出字符串的长度 `ck`。

然后, 用变量 `n` 表示循环次数, 变量 `m` 表示经过多少次循环才刷新一次进度条, 用 `For` 语句进行 `n` 次循环。

每次循环除了模拟要执行的操作(在当前文档输出一个随机数并换行)外, 还要判断循环变量 `k` 能否被 `m` 整除。若 `k` 能被 `m` 整除, 即 `k` 为 `m` 的整数倍, 则求出进度条应有的实心菱形块数量, 从 `sk` 和 `kk` 字符串左右两边分别取出一定数量的字符, 拼成新的字符串用 `p` 表示, 并将 `p` 与变量 `wtm` 的值拼接后显示在系统的状态栏上。语句 `DoEvents` 转让控制权给操作系统, 起到刷新屏幕作用。

最后, 恢复系统状态栏。

运行“显示进度”子程序, 会在当前 Word 文档输出模拟数据, 并在状态栏上动态显示进度条, 如图 5-5 所示。

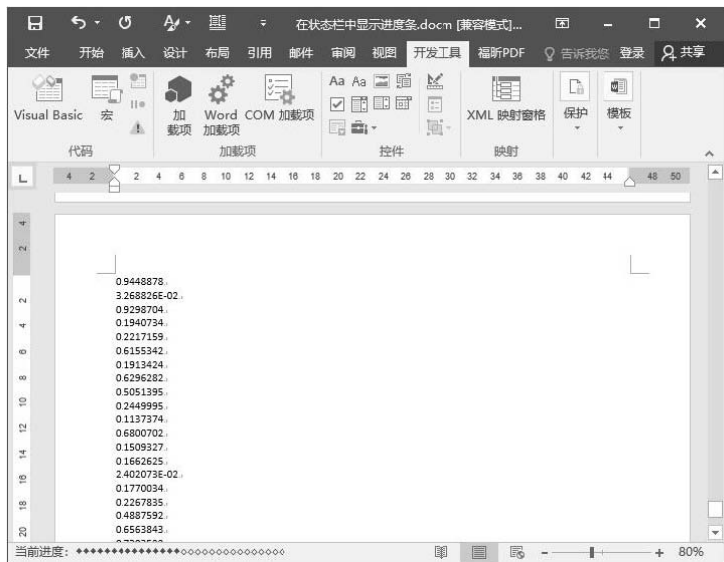


图 5-5 Word 状态栏上的进度条

5.6 哥德巴赫猜想问题

公元 1742 年, 德国数学家哥德巴赫提出了著名的猜想: 任何一个大于或等于 6 的偶数,

都可以表示成两个素数之和。例如， $6=3+3$ 、 $8=3+5$ 、 $10=5+5$ 、 $\cdots$ 、 $100=3+97=11+89=17+83$ ，等等。在这些具体的例子中，可以看出哥德巴赫猜想都是成立的。有人甚至逐一验证了 3300 万以内的所有偶数，竟然没有一个不符合哥德巴赫猜想的。随着计算机技术的发展，数学家们发现哥德巴赫猜想对于更大的数依然成立。可是自然数是无限的，谁知道会不会在某一个足够大的偶数上，突然出现哥德巴赫猜想的反例呢？因此用逐一验证的方法显然不可取。

本节绝无证明哥德巴赫猜想之意，只是出于好奇，通过 VBA 程序，在一定范围内验证哥德巴赫猜想。目的是激发学习兴趣，提高程序设计技巧和应用能力。

1. 编写自定义函数

判断一个数是否为素数，可以用一个自定义函数来实现。

创建一个 Word 文档，保存为“哥德巴赫猜想问题.docm”。按 Alt+F11 快捷键，进入 VB 编辑环境。插入一个模块，编写一个自定义函数 isprime，代码如下：

```
Function isprime(n)
  For k = 2 To Sqr(n)
    If n Mod k = 0 Then
      isprime = False
      Exit Function
    End If
  Next
  isprime = True
End Function
```

上述自定义函数的功能是判断自变量 **n** 是否为素数。如果 **n** 是素数，函数的返回值为 **True**，否则，函数的返回值为 **False**。方法是用 2 到 **Sqr(n)** 之间的所有整数去试除 **n**，如果这些数都不能整除 **n**，则 **n** 是素数，否则 **n** 不是素数。

2. 编写子程序

在模块中编写一个“哥德巴赫猜想”子程序，代码如下：

```
Sub 哥德巴赫猜想()  
    wtm = "当前进度: "  
    kk = "◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇"  
    sk = "◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆"  
    ck = Len(kk)           '进度条长度  
    n = 500                '循环次数  
    m = 1 + n \ ck         '每循环m次，刷新进度条1次  
    i = 2000               '偶数起始值  
    For k = 1 To n        '循环  
        j = 3  
        Do  
            If isprime(j) And isprime(i - j) Then  
                Selection.EndKey Unit:=wdStory      '光标定位到文档尾  
                Selection.TypeText Text:=i & "=" & j & "+" & i - j & Chr(10)  
            Exit Do  
        End If
```