

NOTE AA: GENERATE DIEHARD TEST INPUT FILES

```
Sub BuildDiehardFile()
Dim hn, hpos1, hout, hix, hiy, hiz As Long
Dim xa, xb, xn As Double
'Const refh = 4294967296#
'Const refl = 2147483647#

hpos1 = 1&
hn = 1&

Rem uses Sobel's recommended starting seed
hix = 5&
hix = 11&
hiz = 7&

Open "C:\newtrial.bin" For Binary As 2

With Application.ActiveSheet
Do
Call Generate_WH_FPRN(hix, hiy, hiz, xn) 'used for the equation set
'Calculate 'optional, to ensure new RAND value in cell(1,1)
'xn = .Cells(1, 1).Value 'cell(1,1) has the equation =RAND()
xa = refh * xn
If xa > refl Then
xb = xa - refl
Else
xb = xa
End If
hout = CLng(xb) 'hout has the correct bit sequence of an unsigned long integer
Put #2, hpos1, hout
hpos1 = hpos1 + 4&
hn = hn + 1&
.Cells(1, 2) = hn 'visual track on progress
If hn > 3000000 Then 'for diehard-II, use 3000000, for Diehard-III, use
67000000
Exit Do
End If
Loop
Close
End With
ActiveWorkbook.Save

End Sub
```

The fault here is that the file “C:\newtrial.bin” results in a catastrophically failure in Diehard-II. A detailed examination of the bit patterns shows that the fault appears to be in the VBA module.

Given the following first four bit patterns of hout, (going into the file)

```
0001 1001 1111 0100 0111 1110 0101 0101
0100 1100 0100 0100 0111 1011 0100 0100
0100 0101 1101 0110 1010 1111 1001 1110
0111 1110 1001 0010 0111 0110 1011 0001
```

The bit patterns in the file after closing and reopening are,

```
0111 1110 0101 0101 0000 0000 0000 0011
0111 1011 0100 0100 0000 0000 0000 0011
1010 1111 1001 1110 0000 0000 0000 0011
0111 0110 1001 0010 0000 0000 0000 0011
```

The Diehard-II reports shows a total failure in bits 16-32, indicating that the loss of the input bits is correct.

This is a fault in VBA, incorrectly building a binary file. The instruction “ Put #2, hpos1, hout” inserts the last 4 bytes of hout into the first 4 bytes of the address and puts in the sequence “0000 0000 0000 0011” to fill out the 8 byte number.

When running the above subroutine with stops, hout has the correct long integer value.

A Visual Basic routine was built to properly build a binary file from the respective random number algorithms. It was run in Visual Basic 6.