Example 1: Demonstrating Integer Data Definition

```
TITLE Integer Data Definitions            (File:IntegerDef.asm)
; Examples Demonstrating Integer Data Definition
.686
.MODEL flat, stdcall
.STACK
INCLUDE Irvine32.inc
.data
; ----------------- Byte Values --------------------
byte1  BYTE   'A'   ; 'A' = 65 = 41h
byte2  BYTE    0    ; smallest unsigned byte value
byte3  BYTE    255  ; largest unsigned byte value
byte4  SBYTE   -128 ; smallest signed byte value
byte5  SBYTE   +127 ; largest signed byte value   ;
byte6  BYTE    ?    ; uninitialized
; ----------------- Word Values --------------------
word1  WORD    65535     ; largest unsigned word value
word2  SWORD   -32768    ; smallest signed word value
word3  WORD    ?    ; uninitialized
; --------------- DoubleWord Values -----------------
dword1 DWORD   0FFFFFFFFh     ; largest unsigned value in hex
dword2 SDWORD  -2147483648; smallest signed value in decimal
; --------------- QuadWord Value --------------------
quad1  QWORD   0123456789ABCDEFh
.code
main PROC
; No instructions to execute
     exit
main ENDP
END main
```

Example 2: Multiple Initializers and the DUP Operator

```
TITLE Multiple Initializers        (MultipleInitializers.asm)
; Examples showing multiple initializers and the DUP operator
.686
.MODEL flat, stdcall
.STACK
INCLUDE Irvine32.inc
; ----------------- Byte Values ---------------
.data
list1   BYTE  10, 32, 41h, 00100010b
list2   BYTE  0Ah, 20h, 'A', 22h
array1  BYTE  8 DUP(0)
greeting BYTE "Good  afternoon",0
; ----------------- Word Values --------------------
myList  WORD   1,2,3,4   ; array of words
; --------------- DoubleWord Values -------------
```

```
array2  DWORD   4 DUP(01234567h)
.code
main PROC
; No instructions to execute
     exit
main ENDP
END main
```

Example 3: Symbolic Constants

```
TITLE Symbolic Constants            (File: Constants.asm)
; Demonstration of EQU and = directives
.686
.MODEL flat, stdcall
.STACK
INCLUDE Irvine32.inc
.data
Rows       EQU   3
Cols       EQU   3
Elements   EQU   Rows * Cols
CR         EQU   10
LF         EQU   13
PromptText EQU   <"Press any key to continue ...",CR,LF,0>
matrix     WORD  Elements DUP(0)
prompt     BYTE  PromptText
COUNT = 10h
COUNT = 100h
COUNT = 1000h
COUNT = SIZEOF matrix
.code
main PROC
     exit
main ENDP
END main
```