Marco Martinez
CISP 310
Assignment 2c

Hierarchy Chart

==========

```
3.0 Encryption
        3.1 InputTheString
        3.2 DisplayMessage
        3.3 TranslateBuffer
```

Pseudocode

==========

Main Module

```
Begin
        Define BUFMAX as Constant Integer = 128
        Declare sPrompt as String = “Enter the plain text: “
        Declare sEncrypt as String = “Cipher text: “
        Declare sDecrypt as String = “Decrypted: “
        Declare buffer as Byte[1…BUFMAX]
        Declare bufSize as DoubleWord
        Declare encryptKey as String "ABXmv#7"

        Call InputTheString
        Call TranslateBuffer
        Set edx as sEncrypt Address Start
        Call DisplayMessage(sEncrypt)
        Call TranslateBuffer
        Set edx as sDecrypt Address Start
        Call DisplayMessage(sDecrypt)
end Main
```

InputTheString Module

```
Begin
        Pushad
        Set edx as sPrompt Address Start
        Call WriteString
        Set ecx as BUFMAX
        Set edxa as buffer Address Start
        Call ReadString
        Set bufSize as eax
        Call Crlf
        Popad
Return
```

DisplayMessage Module

```
Begin
        Pushad
        Call WriteString
        Set edx as buffer Address Start
```

```
        Call WriteString
        Call Crlf
        Call Crlf
        Popad
Return
```

TranslateBuffer Module

```
Begin
        Pushad
        Set ecx as bufSiz
        Set esi as 0
        Set edi as 0
L1:
        If(edi >= 7)
                Set edi as 0
        EndIf
        Set al as encryptKey at index edi
        If(buffer at index esi != al)
                Set buffer at index esi as al
        EndIf
        Esi = esi + 1
        Edi = edi + 1
        Loop L1
        Popad
Return
```

Listing File

```
Microsoft (R) Macro Assembler Version 14.15.26732.1           12/03/18 23:55:38
..\..\..\..\Documents\School Work\P310\secondProject\ASM 2c\encryption.asm  Page 1 - 1


                                ;;      Author:         Marco Martinez
                                ;;      Filename:               encryption.asm
                                ;;      Version:                1.0
                                ;;      Description:    Revise the encryption program in Section 6.3.4 in the following manner: Create an encryption
                                ;;                              key consisting of multiple characters. Use this key to encrypt and decrypt the
                                ;;                              plaintext by XORing each character of the key against a corresponding byte in the
                                ;;                              message. Repeat the key as many times as necessary until all plain text bytes are
                                ;;                              translated. Suppose, for example, the key were equal to “ABXmv#7”.
                                ;;      Date:           10/28
                                ;;
                                ;;      Program Change Log
                                ;;      ==================
                                ;;      Name            Date            Description
                                ;;      Marco   10/28   Create baseline for encryption.asm
                                ;;

                                INCLUDE Irvine32.inc
                               C ; Include file for Irvine32.lib             (Irvine32.inc)
```

```
                                C
                                C ;OPTION CASEMAP:NONE        ; optional: make identifiers case-sensitive
                                C
                                C INCLUDE SmallWin.inc        ; MS-Windows prototypes, structures, and constants
                                C .NOLIST
                                C .LIST
                                C
                                C INCLUDE VirtualKeys.inc
                                C ; VirtualKeys.inc
                                C .NOLIST
                                C .LIST
                                C
                                C
                                C .NOLIST
                                C .LIST
                                C
= 00000080                       BUFMAX = 128 ; maximum buffer size

00000000                         .data
00000000 45 6E 74 65 72                 sPrompt BYTE "Enter the plain text: ",0
         20 74 68 65 20
         70 6C 61 69 6E
         20 74 65 78 74
         3A 20 00
00000017 43 69 70 68 65                 sEncrypt BYTE "Cipher text: ",0
         72 20 74 65 78
         74 3A 20 00
00000025 44 65 63 72 79                 sDecrypt BYTE "Decrypted: ",0
         70 74 65 64 3A
         20 00
00000031  00000081 [                    buffer BYTE BUFMAX+1 DUP(0)
          00
         ]
000000B2 00000000                       bufSize DWORD ?
000000B6 41 42 58 6D 76                 encryptKey BYTE "ABXmv#7",0
         23 37 00

00000000                         .code
00000000                         main PROC
00000000  E8 00000025                           call InputTheString ; input the plain text
00000005  E8 00000062                           call TranslateBuffer ; encrypt the buffer
0000000A  BA 00000017 R                 mov edx,OFFSET sEncrypt ; display encrypted message
0000000F  E8 0000003C                           call DisplayMessage
00000014  E8 00000053                           call TranslateBuffer ; decrypt the buffer
00000019  BA 00000025 R                 mov edx,OFFSET sDecrypt ; display decrypted message
0000001E  E8 0000002D                           call DisplayMessage
                                        exit
00000023  6A 00         *       push   +000000000h
00000025  E8 00000000 E   *     call   ExitProcess
0000002A                         main ENDP
```

```
                                ;-----------------------------------------------------
0000002A                        InputTheString PROC
                                ;
                                ; Prompts user for a plaintext string. Saves the string
                                ; and its length.
                                ; Receives: nothing
                                ; Returns: nothing
                                ;-----------------------------------------------------
0000002A  60                            pushad ; save 32-bit registers
0000002B  BA 00000000 R                 mov edx,OFFSET sPrompt ; display a prompt
00000030  E8 00000000 E                 call WriteString
00000035  B9 00000080                           mov ecx,BUFMAX ; maximum character count
0000003A  BA 00000031 R                 mov edx,OFFSET buffer ; point to the buffer
0000003F  E8 00000000 E                 call ReadString ; input the string
00000044  A3 000000B2 R                 mov bufSize,eax ; save the length
00000049  E8 00000000 E                 call Crlf
0000004E  61                            popad
0000004F  C3                            ret
00000050                        InputTheString ENDP

                                ;-----------------------------------------------------
00000050                        DisplayMessage PROC
                                ;
                                ; Displays the encrypted or decrypted message.
                                ; Receives: EDX points to the message
                                ; Returns: nothing
                                ;-----------------------------------------------------
00000050  60                            pushad
00000051  E8 00000000 E                 call WriteString
00000056  BA 00000031 R                 mov edx,OFFSET buffer ; display the buffer
0000005B  E8 00000000 E                 call WriteString
00000060  E8 00000000 E                 call Crlf
00000065  E8 00000000 E                 call Crlf
0000006A  61                            popad
0000006B  C3                            ret
0000006C                        DisplayMessage ENDP

                                ;-----------------------------------------------------
0000006C                        TranslateBuffer PROC
                                ;
                                ; Translates the string by exclusive-ORing each
                                ; byte with the encryption key byte.
                                ; Receives: nothing
                                ; Returns: nothing
                                ;-----------------------------------------------------
0000006C  60                            pushad
0000006D  8B 0D 000000B2 R              mov ecx,bufSize ; loop counter
00000073  BE 00000000                           mov esi,0 ; index 0 in buffer
00000078  BF 00000000                           mov edi,0 ; index 0 in encryptKey
0000007D                        L1:
0000007D  83 FF 06                      cmp edi,6
```

```
00000080  72 05                    jb AvoidReset ; jump if edi is less than 6, sidestepping the reassignment process
00000082  BF 00000000                     mov edi,0     ; reset edi for looping through encryptKey array if it does not equal 7 (0-6)
00000087                     AvoidReset:
00000087  8A 87 000000B6 R         mov al, encryptKey[edi] ; assign encryptionKey byte at index edi to al
0000008D  30 86 00000031 R         xor buffer[esi],al ; translate a byte from al to buffer if they are not equal
00000093  46                       inc esi ; point to next buffer byte
00000094  47                       inc edi ; point to next key byte
00000095  E2 E6                    loop L1
00000097  61                       popad ; reset registers
00000098  C3                       ret ; return
00000099                     TranslateBuffer ENDP
                             END main
```

Structures and Unions:

N a m e	Size Offset	Type
CONSOLE_CURSOR_INFO	00000008	
dwSize	00000000	DWord
bVisible	00000004	DWord
CONSOLE_SCREEN_BUFFER_INFO . . .	00000016	
dwSize	00000000	DWord
dwCursorPosition	00000004	DWord
wAttributes	00000008	Word
srWindow	0000000A	QWord
dwMaximumWindowSize	00000012	DWord
COORD	00000004	
X	00000000	Word
Y	00000002	Word
FILETIME	00000008	
loDateTime	00000000	DWord
hiDateTime	00000004	DWord
FOCUS_EVENT_RECORD	00000004	
bSetFocus	00000000	DWord
FPU_ENVIRON	0000001C	
controlWord	00000000	Word
statusWord	00000004	Word
tagWord	00000008	Word
instrPointerOffset	0000000C	DWord
instrPointerSelector	00000010	DWord
operandPointerOffset	00000014	DWord
operandPointerSelector	00000018	Word
INPUT_RECORD	00000014	
EventType	00000000	Word
Event	00000004	XmmWord
bKeyDown	00000000	DWord
wRepeatCount	00000004	Word
wVirtualKeyCode	00000006	Word
wVirtualScanCode	00000008	Word
uChar	0000000A	Word
UnicodeChar	00000000	Word
AsciiChar	00000000	Byte
dwControlKeyState	0000000C	DWord
dwMousePosition	00000000	DWord

```
  dwButtonState  . . . . . . . .   00000004      DWord
  dwMouseControlKeyState . . . .   00000008      DWord
  dwEventFlags . . . . . . . . .   0000000C      DWord
  dwSize . . . . . . . . . . . .   00000000      DWord
  dwCommandId  . . . . . . . . .   00000000      DWord
  bSetFocus  . . . . . . . . . .   00000000      DWord
KEY_EVENT_RECORD . . . . . . . .   00000010
  bKeyDown . . . . . . . . . . .   00000000      DWord
  wRepeatCount . . . . . . . . .   00000004      Word
  wVirtualKeyCode  . . . . . . .   00000006      Word
  wVirtualScanCode . . . . . . .   00000008      Word
  uChar  . . . . . . . . . . . .   0000000A      Word
  UnicodeChar  . . . . . . . . .   00000000      Word
  AsciiChar  . . . . . . . . . .   00000000      Byte
  dwControlKeyState  . . . . . .   0000000C      DWord
MENU_EVENT_RECORD  . . . . . . .   00000004
  dwCommandId  . . . . . . . . .   00000000      DWord
MOUSE_EVENT_RECORD . . . . . . .   00000010
  dwMousePosition  . . . . . . .   00000000      DWord
  dwButtonState  . . . . . . . .   00000004      DWord
  dwMouseControlKeyState . . . .   00000008      DWord
  dwEventFlags . . . . . . . . .   0000000C      DWord
SMALL_RECT . . . . . . . . . . .   00000008
  Left . . . . . . . . . . . . .   00000000      Word
  Top  . . . . . . . . . . . . .   00000002      Word
  Right  . . . . . . . . . . . .   00000004      Word
  Bottom . . . . . . . . . . . .   00000006      Word
SYSTEMTIME . . . . . . . . . . .   00000010
  wYear  . . . . . . . . . . . .   00000000      Word
  wMonth . . . . . . . . . . . .   00000002      Word
  wDayOfWeek . . . . . . . . . .   00000004      Word
  wDay . . . . . . . . . . . . .   00000006      Word
  wHour  . . . . . . . . . . . .   00000008      Word
  wMinute  . . . . . . . . . . .   0000000A      Word
  wSecond  . . . . . . . . . . .   0000000C      Word
  wMilliseconds  . . . . . . . .   0000000E      Word
WINDOW_BUFFER_SIZE_RECORD  . . .   00000004
  dwSize . . . . . . . . . . . .   00000000      DWord


Segments and Groups:

                N a m e                 Size     Length   Align   Combine Class

FLAT . . . . . . . . . . . . . .   GROUP
STACK  . . . . . . . . . . . . .   32 Bit  00001000 Para         Stack         'STACK'
_DATA  . . . . . . . . . . . . .   32 Bit  000000BE Para         Public  'DATA'
_TEXT  . . . . . . . . . . . . .   32 Bit  00000099 Para         Public  'CODE'
```

```
Procedures, parameters, and locals:

                N a m e                 Type     Value    Attr

CloseFile  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CloseHandle  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Clrscr . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CreateFileA  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CreateOutputFile . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Crlf . . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Delay  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
DisplayMessage . . . . . . . . .  P Near  00000050 _TEXT     Length= 0000001C Public STDCALL
DumpMem  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
DumpRegs . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ExitProcess  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FileTimeToDosDateTime  . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FileTimeToSystemTime . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FlushConsoleInputBuffer  . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FormatMessageA . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetCommandLineA  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetCommandTail . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleCP . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleCursorInfo . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleMode . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetDateTime  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetFileTime  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetKeyState  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetLastError . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetLocalTime . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetMaxXY . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetMseconds  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetNumberOfConsoleInputEvents  .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetProcessHeap . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetStdHandle . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetSystemTime  . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetTextColor . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetTickCount . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Gotoxy . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapAlloc  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapCreate . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapDestroy  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapFree . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapSize . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
InputTheString . . . . . . . . .  P Near  0000002A _TEXT     Length= 00000026 Public STDCALL
IsDigit  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
```

```
LocalFree  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
MessageBoxA  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
MsgBoxAsk  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
MsgBox . . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
OpenInputFile  . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ParseDecimal32 . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ParseInteger32 . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
PeekConsoleInputA  . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Random32 . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
RandomRange  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Randomize  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadChar . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadConsoleA . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadConsoleInputA  . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadDec  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadFile . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadFloat  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadFromFile . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadHex  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadInt  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadKeyFlush . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadKey  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ReadString . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleCursorInfo . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleCursorPosition . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleMode . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleScreenBufferSize . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleTextAttribute  . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleTitleA . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetConsoleWindowInfo . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetFilePointer . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetLocalTime . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SetTextColor . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
ShowFPUStack . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Sleep  . . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
StrLength  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Str_compare  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Str_copy . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Str_length . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Str_trim . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
Str_ucase  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
SystemTimeToFileTime . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
TranslateBuffer  . . . . . . . . .  P Near  0000006C _TEXT      Length= 0000002D Public STDCALL
  L1 . . . . . . . . . . . . . . .  L Near  0000007D _TEXT
  AvoidReset . . . . . . . . . . .  L Near  00000087 _TEXT
WaitMsg  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteBinB  . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteBin . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
```

```
WriteChar  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteConsoleA  . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteConsoleOutputAttribute  . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteConsoleOutputCharacterA . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteDec . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteFile  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteFloat . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteHexB  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteHex . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteInt . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteStackFrameName  . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteStackFrame  . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteString  . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteToFile  . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
WriteWindowsMsg  . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External STDCALL
main . . . . . . . . . . . . . .  P Near  00000000 _TEXT      Length= 0000002A Public STDCALL
printf . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External C
scanf  . . . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External C
wsprintfA  . . . . . . . . . . .  P Near  00000000 FLAT       Length= 00000000 External C


Symbols:

                N a m e                 Type     Value    Attr

@CodeSize  . . . . . . . . . . .  Number  00000000h
@DataSize  . . . . . . . . . . .  Number  00000000h
@Interface . . . . . . . . . . .  Number  00000003h
@Model . . . . . . . . . . . . .  Number  00000007h
@code  . . . . . . . . . . . . .  Text           _TEXT
@data  . . . . . . . . . . . . .  Text           FLAT
@fardata?  . . . . . . . . . . .  Text           FLAT
@fardata . . . . . . . . . . . .  Text           FLAT
@stack . . . . . . . . . . . . .  Text           FLAT
ALT_MASK . . . . . . . . . . . .  Number  00000003h
BUFMAX . . . . . . . . . . . . .  Number  00000080h
CAPSLOCK_ON  . . . . . . . . . .  Number  00000080h
CREATE_ALWAYS  . . . . . . . . .  Number  00000002h
CREATE_NEW . . . . . . . . . . .  Number  00000001h
CTRL_MASK  . . . . . . . . . . .  Number  0000000Ch
CreateFile . . . . . . . . . . .  Text           CreateFileA
DO_NOT_SHARE . . . . . . . . . .  Number  00000000h
ENABLE_ECHO_INPUT  . . . . . . .  Number  00000004h
ENABLE_LINE_INPUT  . . . . . . .  Number  00000002h
ENABLE_MOUSE_INPUT . . . . . . .  Number  00000010h
ENABLE_PROCESSED_INPUT . . . . .  Number  00000001h
ENABLE_PROCESSED_OUTPUT  . . . .  Number  00000001h
ENABLE_WINDOW_INPUT  . . . . . .  Number  00000008h
```

```
ENABLE_WRAP_AT_EOL_OUTPUT  . . .  Number  00000002h
ENHANCED_KEY . . . . . . . . . .  Number  00000100h
FALSE  . . . . . . . . . . . . .  Number  00000000h
FILE_APPEND_DATA . . . . . . . .  Number  00000004h
FILE_ATTRIBUTE_ARCHIVE . . . . .  Number  00000020h
FILE_ATTRIBUTE_COMPRESSED  . . .  Number  00000800h
FILE_ATTRIBUTE_DEVICE  . . . . .  Number  00000040h
FILE_ATTRIBUTE_DIRECTORY . . . .  Number  00000010h
FILE_ATTRIBUTE_ENCRYPTED . . . .  Number  00004000h
FILE_ATTRIBUTE_HIDDEN  . . . . .  Number  00000002h
FILE_ATTRIBUTE_NORMAL  . . . . .  Number  00000080h
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED .     Number  00002000h
FILE_ATTRIBUTE_OFFLINE . . . . .  Number  00001000h
FILE_ATTRIBUTE_READONLY  . . . .  Number  00000001h
FILE_ATTRIBUTE_REPARSE_POINT . .  Number  00000400h
FILE_ATTRIBUTE_SPARSE_FILE . . .  Number  00000200h
FILE_ATTRIBUTE_SYSTEM  . . . . .  Number  00000004h
FILE_ATTRIBUTE_TEMPORARY . . . .  Number  00000100h
FILE_BEGIN . . . . . . . . . . .  Number  00000000h
FILE_CURRENT . . . . . . . . . .  Number  00000001h
FILE_DELETE_CHILD  . . . . . . .  Number  00000040h
FILE_END . . . . . . . . . . . .  Number  00000002h
FILE_READ_DATA . . . . . . . . .  Number  00000001h
FILE_SHARE_DELETE  . . . . . . .  Number  00000004h
FILE_SHARE_READ  . . . . . . . .  Number  00000001h
FILE_SHARE_WRITE . . . . . . . .  Number  00000002h
FILE_WRITE_DATA  . . . . . . . .  Number  00000002h
FOCUS_EVENT  . . . . . . . . . .  Number  00000010h
FORMAT_MESSAGE_ALLOCATE_BUFFER .  Number  00000100h
FORMAT_MESSAGE_FROM_SYSTEM . . .  Number  00001000h
FormatMessage  . . . . . . . . .  Text          FormatMessageA
GENERIC_ALL  . . . . . . . . . .  Number  10000000h
GENERIC_EXECUTE  . . . . . . . .  Number  20000000h
GENERIC_READ . . . . . . . . . .  Number  -80000000h
GENERIC_WRITE  . . . . . . . . .  Number  40000000h
GetCommandLine . . . . . . . . .  Text          GetCommandLineA
HANDLE . . . . . . . . . . . . .  Text          DWORD
HEAP_GENERATE_EXCEPTIONS . . . .  Number  00000004h
HEAP_GROWABLE  . . . . . . . . .  Number  00000002h
HEAP_NO_SERIALIZE  . . . . . . .  Number  00000001h
HEAP_REALLOC_IN_PLACE_ONLY . . .  Number  00000010h
HEAP_ZERO_MEMORY . . . . . . . .  Number  00000008h
IDABORT  . . . . . . . . . . . .  Number  00000003h
IDCANCEL . . . . . . . . . . . .  Number  00000002h
IDCLOSE  . . . . . . . . . . . .  Number  00000008h
IDCONTINUE . . . . . . . . . . .  Number  0000000Bh
IDHELP . . . . . . . . . . . . .  Number  00000009h
IDIGNORE . . . . . . . . . . . .  Number  00000005h
```

```
IDNO . . . . . .  Number  00000007h
IDOK . . . . . .  Number  00000001h
IDRETRY  . . . . . .  Number  00000004h
IDTIMEOUT  . . . . . .  Number  00007D00h
IDTRYAGAIN . . . . . .  Number  0000000Ah
IDYES  . . . . . .  Number  00000006h
INVALID_HANDLE_VALUE . . . . . .  Number  -00000001h
KBDOWN_FLAG  . . . . . .  Number  00000001h
KEY_EVENT  . . . . . .  Number  00000001h
KEY_MASKS  . . . . . .  Number  0000001Fh
LEFT_ALT_PRESSED . . . . . .  Number  00000002h
LEFT_CTRL_PRESSED  . . . . . .  Number  00000008h
MB_ABORTRETRYIGNORE  . . . . . .  Number  00000002h
MB_APPLMODAL . . . . . .  Number  00000000h
MB_CANCELTRYCONTINUE . . . . . .  Number  00000006h
MB_DEFBUTTON1  . . . . . .  Number  00000000h
MB_DEFBUTTON2  . . . . . .  Number  00000100h
MB_DEFBUTTON3  . . . . . .  Number  00000200h
MB_DEFBUTTON4  . . . . . .  Number  00000300h
MB_HELP  . . . . . .  Number  00004000h
MB_ICONASTERISK  . . . . . .  Number  00000040h
MB_ICONERROR . . . . . .  Number  00000010h
MB_ICONEXCLAMATION . . . . . .  Number  00000030h
MB_ICONHAND  . . . . . .  Number  00000010h
MB_ICONINFORMATION . . . . . .  Number  00000040h
MB_ICONQUESTION  . . . . . .  Number  00000020h
MB_ICONSTOP  . . . . . .  Number  00000010h
MB_ICONWARNING . . . . . .  Number  00000030h
MB_OKCANCEL  . . . . . .  Number  00000001h
MB_OK  . . . . . .  Number  00000000h
MB_RETRYCANCEL . . . . . .  Number  00000005h
MB_SYSTEMMODAL . . . . . .  Number  00001000h
MB_TASKMODAL . . . . . .  Number  00002000h
MB_USERICON  . . . . . .  Number  00000080h
MB_YESNOCANCEL . . . . . .  Number  00000003h
MB_YESNO . . . . . .  Number  00000004h
MENU_EVENT . . . . . .  Number  00000008h
MOUSE_EVENT  . . . . . .  Number  00000002h
MessageBox . . . . . .  Text          MessageBoxA
NULL . . . . . .  Number  00000000h
NUMLOCK_ON . . . . . .  Number  00000020h
OPEN_ALWAYS  . . . . . .  Number  00000004h
OPEN_EXISTING  . . . . . .  Number  00000003h
PeekConsoleInput . . . . . .  Text          PeekConsoleInputA
RIGHT_ALT_PRESSED  . . . . . .  Number  00000001h
RIGHT_CTRL_PRESSED . . . . . .  Number  00000004h
ReadConsoleInput . . . . . .  Text          ReadConsoleInputA
ReadConsole  . . . . . .  Text          ReadConsoleA
```

```
SCROLLLOCK_ON  . . . . . .  Number  00000040h
SHIFT_MASK . . . . . . . .  Number  00000010h
SHIFT_PRESSED  . . . . . .  Number  00000010h
STD_ERROR_HANDLE . . . . .  Number  -0000000Ch
STD_INPUT_HANDLE . . . . .  Number  -0000000Ah
STD_OUTPUT_HANDLE  . . . .  Number  -0000000Bh
SetConsoleTitle  . . . . .  Text          SetConsoleTitleA
TAB  . . . . . . . . . . .  Number  00000009h
TRUE . . . . . . . . . . .  Number  00000001h
TRUNCATE_EXISTING  . . . .  Number  00000005h
VK_11  . . . . . . . . . .  Number  000000BDh
VK_12  . . . . . . . . . .  Number  000000BBh
VK_ADD . . . . . . . . . .  Number  0000006Bh
VK_BACK  . . . . . . . . .  Number  00000008h
VK_CANCEL  . . . . . . . .  Number  00000003h
VK_CAPITAL . . . . . . . .  Number  00000014h
VK_CLEAR . . . . . . . . .  Number  0000000Ch
VK_CONTROL . . . . . . . .  Number  00000011h
VK_DECIMAL . . . . . . . .  Number  0000006Eh
VK_DELETE  . . . . . . . .  Number  0000002Eh
VK_DIVIDE  . . . . . . . .  Number  0000006Fh
VK_DOWN  . . . . . . . . .  Number  00000028h
VK_END . . . . . . . . . .  Number  00000023h
VK_ESCAPE  . . . . . . . .  Number  0000001Bh
VK_EXECUTE . . . . . . . .  Number  0000002Bh
VK_F10 . . . . . . . . . .  Number  00000079h
VK_F11 . . . . . . . . . .  Number  0000007Ah
VK_F12 . . . . . . . . . .  Number  0000007Bh
VK_F13 . . . . . . . . . .  Number  0000007Ch
VK_F14 . . . . . . . . . .  Number  0000007Dh
VK_F15 . . . . . . . . . .  Number  0000007Eh
VK_F16 . . . . . . . . . .  Number  0000007Fh
VK_F17 . . . . . . . . . .  Number  00000080h
VK_F18 . . . . . . . . . .  Number  00000081h
VK_F19 . . . . . . . . . .  Number  00000082h
VK_F1  . . . . . . . . . .  Number  00000070h
VK_F20 . . . . . . . . . .  Number  00000083h
VK_F21 . . . . . . . . . .  Number  00000084h
VK_F22 . . . . . . . . . .  Number  00000085h
VK_F23 . . . . . . . . . .  Number  00000086h
VK_F24 . . . . . . . . . .  Number  00000087h
VK_F2  . . . . . . . . . .  Number  00000071h
VK_F3  . . . . . . . . . .  Number  00000072h
VK_F4  . . . . . . . . . .  Number  00000073h
VK_F5  . . . . . . . . . .  Number  00000074h
VK_F6  . . . . . . . . . .  Number  00000075h
VK_F7  . . . . . . . . . .  Number  00000076h
VK_F8  . . . . . . . . . .  Number  00000077h
```

```
VK_F9  . . . . . .  Number  00000078h
VK_HELP  . . . . . .  Number  0000002Fh
VK_HOME  . . . . . .  Number  00000024h
VK_INSERT  . . . . . .  Number  0000002Dh
VK_LBUTTON . . . . . .  Number  00000001h
VK_LCONTROL  . . . . . .  Number  000000A2h
VK_LEFT  . . . . . .  Number  00000025h
VK_LMENU . . . . . .  Number  000000A4h
VK_LSHIFT  . . . . . .  Number  000000A0h
VK_MENU  . . . . . .  Number  00000012h
VK_MULTIPLY  . . . . . .  Number  0000006Ah
VK_NEXT  . . . . . .  Number  00000022h
VK_NUMLOCK . . . . . .  Number  00000090h
VK_NUMPAD0 . . . . . .  Number  00000060h
VK_NUMPAD1 . . . . . .  Number  00000061h
VK_NUMPAD2 . . . . . .  Number  00000062h
VK_NUMPAD3 . . . . . .  Number  00000063h
VK_NUMPAD4 . . . . . .  Number  00000064h
VK_NUMPAD5 . . . . . .  Number  00000065h
VK_NUMPAD6 . . . . . .  Number  00000066h
VK_NUMPAD7 . . . . . .  Number  00000067h
VK_NUMPAD8 . . . . . .  Number  00000068h
VK_NUMPAD9 . . . . . .  Number  00000069h
VK_PAUSE . . . . . .  Number  00000013h
VK_PRINT . . . . . .  Number  0000002Ah
VK_PRIOR . . . . . .  Number  00000021h
VK_RBUTTON . . . . . .  Number  00000002h
VK_RCONTROL  . . . . . .  Number  000000A3h
VK_RETURN  . . . . . .  Number  0000000Dh
VK_RIGHT . . . . . .  Number  00000027h
VK_RMENU . . . . . .  Number  000000A5h
VK_RSHIFT  . . . . . .  Number  000000A1h
VK_SCROLL  . . . . . .  Number  00000091h
VK_SEPARATER . . . . . .  Number  0000006Ch
VK_SHIFT . . . . . .  Number  00000010h
VK_SNAPSHOT  . . . . . .  Number  0000002Ch
VK_SPACE . . . . . .  Number  00000020h
VK_SUBTRACT  . . . . . .  Number  0000006Dh
VK_TAB . . . . . .  Number  00000009h
VK_UP  . . . . . .  Number  00000026h
WINDOW_BUFFER_SIZE_EVENT . . . .  Number  00000004h
WriteConsoleOutputCharacter  . .  Text          WriteConsoleOutputCharacterA
WriteConsole . . . . . .  Text          WriteConsoleA
black  . . . . . .  Number  00000000h
blue . . . . . .  Number  00000001h
brown  . . . . . .  Number  00000006h
bufSize  . . . . . .  DWord   000000B2 _DATA
buffer . . . . . .  Byte    00000031 _DATA
```

```
cyan . . . . . .  Number  00000003h
encryptKey . . . . . . . . . .  Byte    000000B6 _DATA
exit . . . . . . . . . . . . . .  Text          INVOKE ExitProcess,0
gray . . . . . . . . . . . . . .  Number  00000008h
green  . . . . . . . . . . . . .  Number  00000002h
lightBlue  . . . . . . . . . . .  Number  00000009h
lightCyan  . . . . . . . . . . .  Number  0000000Bh
lightGray  . . . . . . . . . . .  Number  00000007h
lightGreen . . . . . . . . . . .  Number  0000000Ah
lightMagenta . . . . . . . . . .  Number  0000000Dh
lightRed . . . . . . . . . . . .  Number  0000000Ch
magenta  . . . . . . . . . . . .  Number  00000005h
red  . . . . . . . . . . . . . .  Number  00000004h
sDecrypt . . . . . . . . . . . .  Byte    00000025 _DATA
sEncrypt . . . . . . . . . . . .  Byte    00000017 _DATA
sPrompt  . . . . . . . . . . . .  Byte    00000000 _DATA
white  . . . . . . . . . . . . .  Number  0000000Fh
wsprintf . . . . . . . . . . . .  Text          wsprintfA
yellow . . . . . . . . . . . . .  Number  0000000Eh

          0 Warnings
          0 Errors
```

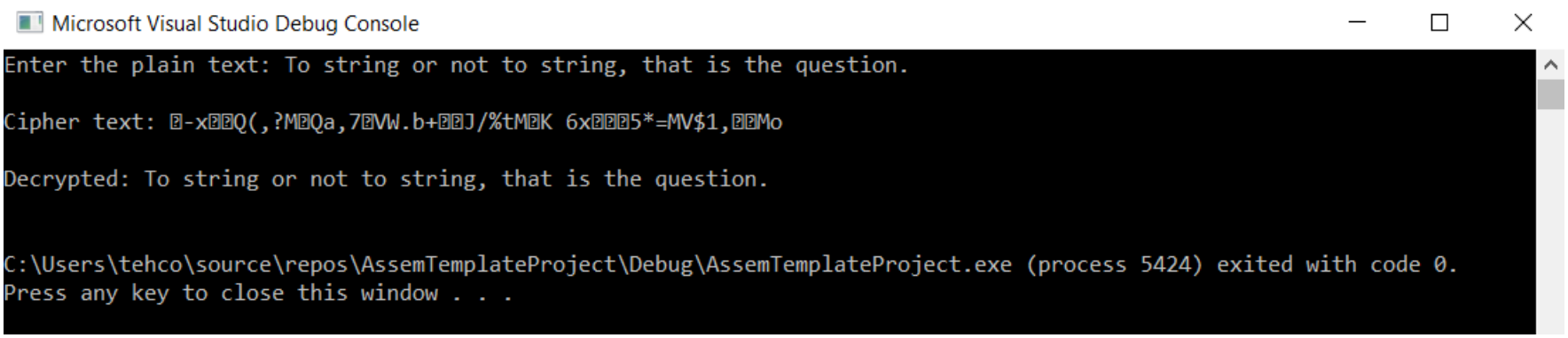
Source Code

```
;;      Author:Marco Martinez
;;      Filename:      encryption.asm
;;      Version:       1.0
;;      Description:   Revise the encryption program in Section 6.3.4 in the following manner: Create an
;;                     encryption key consisting of multiple characters. Use this key to encrypt and
;;                     decrypt the plaintext by XORing each character of the key against a
;;                     corresponding byte in the message. Repeat the key as many times as necessary until
;;                     all plain text bytes are translated. Suppose, for example, the key were equal to
;;                     “ABXmv#7”.
;;      Date:          10/28
;;
;;      Program Change Log
;;      ==================
;;      Name           Date           Description
;;      Marco   10/28  Create baseline for encryption.asm
;;

INCLUDE Irvine32.inc
BUFMAX = 128 ; maximum buffer size

.data
        sPrompt BYTE "Enter the plain text:",0
        sEncrypt BYTE "Cipher text: ",0
        sDecrypt BYTE "Decrypted: ",0
```

```
        buffer BYTE BUFMAX+1 DUP(0)
        bufSize DWORD ?
        encryptKey BYTE "ABXmv#7",0

.code
main PROC
        call InputTheString ; input the plain text
        call TranslateBuffer ; encrypt the buffer
        mov edx,OFFSET sEncrypt ; display encrypted message
        call DisplayMessage
        call TranslateBuffer ; decrypt the buffer
        mov edx,OFFSET sDecrypt ; display decrypted message
        call DisplayMessage
        exit
main ENDP

;-----------------------------------------------------
InputTheString PROC
;
; Prompts user for a plaintext string. Saves the string
; and its length.
; Receives: nothing
; Returns: nothing
;-----------------------------------------------------
        pushad ; save 32-bit registers
        mov edx,OFFSET sPrompt ; display a prompt
        call WriteString
        mov ecx,BUFMAX ; maximum character count
        mov edx,OFFSET buffer ; point to the buffer
        call ReadString ; input the string
        mov bufSize,eax ; save the length
        call Crlf
        popad
        ret
InputTheString ENDP

;-----------------------------------------------------
DisplayMessage PROC
;
; Displays the encrypted or decrypted message.
; Receives: EDX points to the message
; Returns: nothing
;-----------------------------------------------------
        pushad
        call WriteString
        mov edx,OFFSET buffer ; display the buffer
        call WriteString
        call Crlf
        call Crlf
        popad
        ret
```

```
DisplayMessage ENDP

;-----------------------------------------------------
TranslateBuffer PROC
;
; Translates the string by exclusive-ORing each
; byte with the encryption key byte.
; Receives: nothing
; Returns: nothing
;-----------------------------------------------------
        pushad
        mov ecx,bufSize ; loop counter
        mov esi,0 ; index 0 in buffer
        mov edi,0 ; index 0 in encryptKey
L1:
        cmp edi,6
        jb AvoidReset ; jump if edi is less than 6, sidestepping the reassignment process
        mov edi,0      ; reset edi for looping through encryptKey array if it does not equal 7 (0-6)
AvoidReset:
        mov al, encryptKey[edi] ; assign encryptionKey byte at index edi to al
        xor buffer[esi],al ; translate a byte from al to buffer if they are not equal
        inc esi ; point to next buffer byte
        inc edi ; point to next key byte
        loop L1
        popad ; reset registers
        ret ; return
TranslateBuffer ENDP
        END main
```

Screenshot



Hierarchy Chart

```
==========================================================================================
3.0 gradeCalc
        3.1 DetermineInt
        3.2 DisplayGrade
```

3.3 Calc

Pseudocode

==========

Main Module

```
Begin
	Define MIN_A as Constant Integer = 90
	Define MIN_B as Constant Integer = 80
	Define MIN_C as Constant Integer = 70
	Define MIN_D as Constant Integer = 60
	Define MIN_F as Constant Integer = 50
	Define MIN_BOUND as Constant Integer = 0
	Define MAX_BOUND as Constant Integer = 100
	Declare gradeA as String = "A"
	Declare gradeB as String = "B"
	Declare gradeC as String = "C"
	Declare gradeD as String = "D"
	Declare gradeF as String = "F"
	Declare gradeFinal as String
	Declare outOfBounds as String = "Value out of bounds."
	Declare msg as String = " equates to ",0
	Declare counter = 0

	Call Randomize
	Set ecx as 10
	L1:
		Call DetermineInt
		Call Calc
		Call DisplayGrade
		Loop L1
	End while
End main
```

DetermineInt Module

```
Begin
	Set eax as 50
	Call RandomRange
	Set eax as eax + 51
	Set edx as eax
	Call WriteDec
Return
```

DisplayGrade Module

```
Begin
	Set edx as 0
	Set edx as msg address start
	Call WriteString
	Set gradeFinal as al
```

```
        Set edx as gradeFinal address start
        Call WriteString
        Call Crlf
Return
```

GradeCalc Module

```
Begin
        If(eax > MAX_BOUND OR eax < MIN_BOUND)
                Set edx as outOfBounds address start
                call WriteString
                call Crlf
                Return
        End if
        If(eax >= MIN_A)
                Set al as gradeA
                Return
        Else If(eax >= MIN_B)
                Set al as gradeB
                Return
        Else If(eax >= MIN_C)
                Set al as gradeC
                Return
        Else If(eax >= MIN_D)
                Set al as gradeD
                Return
        Else If(eax >= MIN_F)
                Set al as gradeF
                Return
        EndIf
        EndIf
        EndIf
        EndIF
        EndIf
Return
```

Listing File

```
Microsoft (R) Macro Assembler Version 14.15.26732.1        12/04/18 00:26:21
..\..\..\..\Documents\School Work\P310\secondProject\ASM 2c\gradeCalc.asm  Page 1 - 1


                        ;;      Author:             Marco Martinez
                        ;;      Filename:           gradeCalc.asm
                        ;;      Version:            1.0
                        ;;      Description:  Create a procedure named CalcGrade that receives an integer value between 0 and 100, and
                        ;;      returns a single capital letter in the AL register. Preserve all other register values between
                        ;;      calls to the procedure. The letter returned by the procedure should be
                        ;;      according to the following ranges: 90 - 100 = A; 80 - 89 = B; 70 - 79 = C, 60 - 69 = D, 0 - 59 = F
```

```
                         ;;     Write a test program that generates 10 random integers between 50 and 100, inclusive. Each
                         ;;     time an integer is generated, pass it to the CalcGrade procedure. You can test your program
                         ;;     using a debugger, or if you prefer to use the book's library, you can display each integer and its
                         ;;     corresponding letter grade. (The Irvine32 library is required for this solution program because it
                         ;;     uses the RandomRange procedure.)
                         ;;
                         ;;     Date:         10/28
                         ;;
                         ;;     Program Change Log
                         ;;     ==================
                         ;;     Name          Date          Description
                         ;;     Marco  10/28  Create baseline for gradeCalc.asm
                         ;;

                         INCLUDE Irvine32.inc
                        C ; Include file for Irvine32.lib             (Irvine32.inc)
                        C
                        C ;OPTION CASEMAP:NONE             ; optional: make identifiers case-sensitive
                        C
                        C INCLUDE SmallWin.inc             ; MS-Windows prototypes, structures, and constants
                        C .NOLIST
                        C .LIST
                        C
                        C INCLUDE VirtualKeys.inc
                        C ; VirtualKeys.inc
                        C .NOLIST
                        C .LIST
                        C
                        C
                        C .NOLIST
                        C .LIST
                        C

00000000                 .data
= 0000005A                     MIN_A EQU 90
= 00000050                     MIN_B EQU 80
= 00000046                     MIN_C EQU 70
= 0000003C                     MIN_D EQU 60
= 00000032                     MIN_F EQU 50
= 00000000                     MIN_BOUND EQU 0
= 00000064                     MAX_BOUND EQU 100
00000000 41 00                      gradeA BYTE "A",0
00000002 42 00                      gradeB BYTE "B",0
00000004 43 00                      gradeC BYTE "C",0
00000006 44 00                        gradeD BYTE "D",0
00000008 46 00                      gradeF BYTE "F",0
0000000A 00 00                        gradeFinal BYTE ?,0
0000000C 56 61 6C 75 65        outOfBounds BYTE "Value out of bounds.",0
```

```
         20 6F 75 74 20
         6F 66 20 62 6F
         75 6E 64 73 2E
         00
00000021 20 65 71 75 61          msg BYTE " equates to ",0
         74 65 73 20 74
         6F 20 00
0000002E 00000000                value DWORD ?

00000000                  .code
00000000                  main PROC

00000000  E8 00000000 E          call Randomize
00000005  B9 0000000A                   mov ecx, 10
0000000A                  L1:
0000000A  E8 00000013                   call DetermineInt
0000000F  E8 00000047                   call GradeCalc
00000014  E8 0000001E                   call DisplayGrade
00000019  E2 EF                  loop L1

                                 exit
0000001B  6A 00        *       push   +000000000h
0000001D  E8 00000000 E   *          call   ExitProcess
00000022                  main ENDP

                          ;-----------------------------------------------------
00000022                  DetermineInt PROC
                          ;
                          ; Determine the integers to be used for grade calculation (random)
                          ; Receives: nothing
                          ; Returns: nothing
                          ;-----------------------------------------------------
00000022  B8 00000032                   mov eax,50
00000027  E8 00000000 E        call RandomRange
0000002C  83 C0 33               add eax,51
0000002F  8B D0                  mov edx,eax
00000031  E8 00000000 E          Call WriteDec
00000036  C3                     ret
00000037                  DetermineInt ENDP

                          ;-----------------------------------------------------
00000037                  DisplayGrade PROC
                          ;
                          ; Displays the grade for each integer
                          ; Receives: nothing
                          ; Returns: nothing
                          ;-----------------------------------------------------
00000037  BA 00000000                   mov edx,0
```

```
0000003C  BA 00000021 R          mov edx,OFFSET msg
00000041  E8 00000000 E          call WriteString
00000046  A2 0000000A R        mov gradeFinal,al
0000004B  BA 0000000A R        mov edx,OFFSET gradeFinal
00000050  E8 00000000 E        call WriteString
00000055  E8 00000000 E        call Crlf
0000005A  C3                     ret
0000005B                  DisplayGrade ENDP

                          ;-----------------------------------------------------
0000005B                  GradeCalc PROC
                          ;
                          ; Calculates which grade the integer represents
                          ; Receives: nothing
                          ; Returns: nothing
                          ;-----------------------------------------------------
0000005B  83 F8 64               cmp eax,MAX_BOUND
0000005E  77 41                  ja OutBounds
00000060  83 F8 00               cmp eax,MIN_BOUND
00000063  72 3C                  jb OutBounds
00000065  83 F8 5A               cmp eax,MIN_A
00000068  73 14                  jae EarnedA
0000006A  83 F8 50               cmp eax,MIN_B
0000006D  73 16                  jae EarnedB
0000006F  83 F8 46               cmp eax,MIN_C
00000072  73 18                  jae EarnedC
00000074  83 F8 3C               cmp eax,MIN_D
00000077  73 1A                  jae EarnedD
00000079  83 F8 32               cmp eax,MIN_F
0000007C  73 1C                  jae EarnedF
0000007E                  EarnedA:
0000007E  A0 00000000 R          mov al,gradeA
00000083  EB 2B                  jmp Stop
00000085                  EarnedB:
00000085  A0 00000002 R          mov al,gradeB
0000008A  EB 24                  jmp Stop
0000008C                  EarnedC:
0000008C  A0 00000004 R          mov al,gradeC
00000091  EB 1D                  jmp Stop
00000093                  EarnedD:
00000093  A0 00000006 R          mov al,gradeD
00000098  EB 16                  jmp Stop
0000009A                  EarnedF:
0000009A  A0 00000008 R          mov al,gradeF
0000009F  EB 0F                  jmp Stop
000000A1                  OutBounds:
000000A1  BA 0000000C R          mov edx,OFFSET outOfBounds
000000A6  E8 00000000 E          call WriteString
```

```
000000AB  E8 00000000 E          call Crlf
000000B0                  Stop:
000000B0  C3                     ret
000000B1                  GradeCalc ENDP
                          END main
```

Structures and Unions:

N a m e	Size Offset	Type
CONSOLE_CURSOR_INFO	00000008	
dwSize	00000000	DWord
bVisible	00000004	DWord
CONSOLE_SCREEN_BUFFER_INFO . . .	00000016	
dwSize	00000000	DWord
dwCursorPosition	00000004	DWord
wAttributes	00000008	Word
srWindow	0000000A	QWord
dwMaximumWindowSize	00000012	DWord
COORD	00000004	
X	00000000	Word
Y	00000002	Word
FILETIME	00000008	
loDateTime	00000000	DWord
hiDateTime	00000004	DWord
FOCUS_EVENT_RECORD	00000004	
bSetFocus	00000000	DWord
FPU_ENVIRON	0000001C	
controlWord	00000000	Word
statusWord	00000004	Word
tagWord	00000008	Word
instrPointerOffset	0000000C	DWord
instrPointerSelector	00000010	DWord
operandPointerOffset	00000014	DWord
operandPointerSelector	00000018	Word
INPUT_RECORD	00000014	
EventType	00000000	Word
Event	00000004	XmmWord
bKeyDown	00000000	DWord
wRepeatCount	00000004	Word
wVirtualKeyCode	00000006	Word
wVirtualScanCode	00000008	Word
uChar	0000000A	Word
UnicodeChar	00000000	Word
AsciiChar	00000000	Byte
dwControlKeyState	0000000C	DWord
dwMousePosition	00000000	DWord

```
  dwButtonState  . . . . . . . .   00000004      DWord
  dwMouseControlKeyState . . . .   00000008      DWord
  dwEventFlags . . . . . . . . .   0000000C      DWord
  dwSize . . . . . . . . . . . .   00000000      DWord
  dwCommandId  . . . . . . . . .   00000000      DWord
  bSetFocus  . . . . . . . . . .   00000000      DWord
KEY_EVENT_RECORD . . . . . . . .   00000010
  bKeyDown . . . . . . . . . . .   00000000      DWord
  wRepeatCount . . . . . . . . .   00000004      Word
  wVirtualKeyCode  . . . . . . .   00000006      Word
  wVirtualScanCode . . . . . . .   00000008      Word
  uChar  . . . . . . . . . . . .   0000000A      Word
  UnicodeChar  . . . . . . . . .   00000000      Word
  AsciiChar  . . . . . . . . . .   00000000      Byte
  dwControlKeyState  . . . . . .   0000000C      DWord
MENU_EVENT_RECORD  . . . . . . .   00000004
  dwCommandId  . . . . . . . . .   00000000      DWord
MOUSE_EVENT_RECORD . . . . . . .   00000010
  dwMousePosition  . . . . . . .   00000000      DWord
  dwButtonState  . . . . . . . .   00000004      DWord
  dwMouseControlKeyState . . . .   00000008      DWord
  dwEventFlags . . . . . . . . .   0000000C      DWord
SMALL_RECT . . . . . . . . . . .   00000008
  Left . . . . . . . . . . . . .   00000000      Word
  Top  . . . . . . . . . . . . .   00000002      Word
  Right  . . . . . . . . . . . .   00000004      Word
  Bottom . . . . . . . . . . . .   00000006      Word
SYSTEMTIME . . . . . . . . . . .   00000010
  wYear  . . . . . . . . . . . .   00000000      Word
  wMonth . . . . . . . . . . . .   00000002      Word
  wDayOfWeek . . . . . . . . . .   00000004      Word
  wDay . . . . . . . . . . . . .   00000006      Word
  wHour  . . . . . . . . . . . .   00000008      Word
  wMinute  . . . . . . . . . . .   0000000A      Word
  wSecond  . . . . . . . . . . .   0000000C      Word
  wMilliseconds  . . . . . . . .   0000000E      Word
WINDOW_BUFFER_SIZE_RECORD  . . .   00000004
  dwSize . . . . . . . . . . . .   00000000      DWord


Segments and Groups:

                N a m e                 Size     Length   Align   Combine Class

FLAT . . . . . . . . . . . . . .   GROUP
STACK  . . . . . . . . . . . . .   32 Bit  00001000 Para         Stack         'STACK'
_DATA  . . . . . . . . . . . . .   32 Bit  00000032 Para         Public  'DATA'
_TEXT  . . . . . . . . . . . . .   32 Bit  000000B1 Para         Public  'CODE'
```

```
Procedures, parameters, and locals:

                N a m e                 Type     Value    Attr

CloseFile  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CloseHandle  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Clrscr . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CreateFileA  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
CreateOutputFile . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Crlf . . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Delay  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
DetermineInt . . . . . . . . . .  P Near  00000022 _TEXT     Length= 00000015 Public STDCALL
DisplayGrade . . . . . . . . . .  P Near  00000037 _TEXT     Length= 00000024 Public STDCALL
DumpMem  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
DumpRegs . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ExitProcess  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FileTimeToDosDateTime  . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FileTimeToSystemTime . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FlushConsoleInputBuffer  . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
FormatMessageA . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetCommandLineA  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetCommandTail . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleCP . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleCursorInfo . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleMode . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetDateTime  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetFileTime  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetKeyState  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetLastError . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetLocalTime . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetMaxXY . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetMseconds  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetNumberOfConsoleInputEvents  .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetProcessHeap . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetStdHandle . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetSystemTime  . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetTextColor . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GetTickCount . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Gotoxy . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
GradeCalc  . . . . . . . . . . .  P Near  0000005B _TEXT     Length= 00000056 Public STDCALL
  EarnedA  . . . . . . . . . . .  L Near  0000007E _TEXT
  EarnedB  . . . . . . . . . . .  L Near  00000085 _TEXT
  EarnedC  . . . . . . . . . . .  L Near  0000008C _TEXT
  EarnedD  . . . . . . . . . . .  L Near  00000093 _TEXT
  EarnedF  . . . . . . . . . . .  L Near  0000009A _TEXT
```

```
  OutBounds  . . . . . . . . . . .  L Near  000000A1 _TEXT
  Stop . . . . . . . . . . . . . .  L Near  000000B0 _TEXT
HeapAlloc  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapCreate . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapDestroy  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapFree . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
HeapSize . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
IsDigit  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
LocalFree  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
MessageBoxA  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
MsgBoxAsk  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
MsgBox . . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
OpenInputFile  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ParseDecimal32 . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ParseInteger32 . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
PeekConsoleInputA  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Random32 . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
RandomRange  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Randomize  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadChar . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadConsoleA . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadConsoleInputA  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadDec  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadFile . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadFloat  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadFromFile . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadHex  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadInt  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadKeyFlush . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadKey  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ReadString . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleCursorInfo . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleCursorPosition . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleMode . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleScreenBufferSize . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleTextAttribute  . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleTitleA . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetConsoleWindowInfo . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetFilePointer . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetLocalTime . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SetTextColor . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
ShowFPUStack . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Sleep  . . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
StrLength  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Str_compare  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Str_copy . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Str_length . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
Str_trim . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
```

```
Str_ucase  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
SystemTimeToFileTime . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WaitMsg  . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteBinB  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteBin . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteChar  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteConsoleA  . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteConsoleOutputAttribute  . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteConsoleOutputCharacterA . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteDec . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteFile  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteFloat . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteHexB  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteHex . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteInt . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteStackFrameName  . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteStackFrame  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteString  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteToFile  . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
WriteWindowsMsg  . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External STDCALL
main . . . . . . . . . . . . . .  P Near  00000000 _TEXT     Length= 00000022 Public STDCALL
  L1 . . . . . . . . . . . . . .  L Near  0000000A _TEXT
printf . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External C
scanf  . . . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External C
wsprintfA  . . . . . . . . . . .  P Near  00000000 FLAT      Length= 00000000 External C


Symbols:

                N a m e                 Type     Value    Attr

@CodeSize  . . . . . . . . . . .  Number  00000000h
@DataSize  . . . . . . . . . . .  Number  00000000h
@Interface . . . . . . . . . . .  Number  00000003h
@Model . . . . . . . . . . . . .  Number  00000007h
@code  . . . . . . . . . . . . .  Text           _TEXT
@data  . . . . . . . . . . . . .  Text           FLAT
@fardata?  . . . . . . . . . . .  Text           FLAT
@fardata . . . . . . . . . . . .  Text           FLAT
@stack . . . . . . . . . . . . .  Text           FLAT
ALT_MASK . . . . . . . . . . . .  Number  00000003h
CAPSLOCK_ON  . . . . . . . . . .  Number  00000080h
CREATE_ALWAYS  . . . . . . . . .  Number  00000002h
CREATE_NEW . . . . . . . . . . .  Number  00000001h
CTRL_MASK  . . . . . . . . . . .  Number  0000000Ch
CreateFile . . . . . . . . . . .  Text           CreateFileA
DO_NOT_SHARE . . . . . . . . . .  Number  00000000h
ENABLE_ECHO_INPUT  . . . . . . .  Number  00000004h
```

```
ENABLE_LINE_INPUT  . . . . . . .  Number  00000002h
ENABLE_MOUSE_INPUT . . . . . . .  Number  00000010h
ENABLE_PROCESSED_INPUT . . . . .  Number  00000001h
ENABLE_PROCESSED_OUTPUT  . . . .  Number  00000001h
ENABLE_WINDOW_INPUT  . . . . . .  Number  00000008h
ENABLE_WRAP_AT_EOL_OUTPUT  . . .  Number  00000002h
ENHANCED_KEY . . . . . . . . . .  Number  00000100h
FALSE  . . . . . . . . . . . . .  Number  00000000h
FILE_APPEND_DATA . . . . . . . .  Number  00000004h
FILE_ATTRIBUTE_ARCHIVE . . . . .  Number  00000020h
FILE_ATTRIBUTE_COMPRESSED  . . .  Number  00000800h
FILE_ATTRIBUTE_DEVICE  . . . . .  Number  00000040h
FILE_ATTRIBUTE_DIRECTORY . . . .  Number  00000010h
FILE_ATTRIBUTE_ENCRYPTED . . . .  Number  00004000h
FILE_ATTRIBUTE_HIDDEN  . . . . .  Number  00000002h
FILE_ATTRIBUTE_NORMAL  . . . . .  Number  00000080h
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED .      Number  00002000h
FILE_ATTRIBUTE_OFFLINE . . . . .  Number  00001000h
FILE_ATTRIBUTE_READONLY  . . . .  Number  00000001h
FILE_ATTRIBUTE_REPARSE_POINT . .  Number  00000400h
FILE_ATTRIBUTE_SPARSE_FILE . . .  Number  00000200h
FILE_ATTRIBUTE_SYSTEM  . . . . .  Number  00000004h
FILE_ATTRIBUTE_TEMPORARY . . . .  Number  00000100h
FILE_BEGIN . . . . . . . . . . .  Number  00000000h
FILE_CURRENT . . . . . . . . . .  Number  00000001h
FILE_DELETE_CHILD  . . . . . . .  Number  00000040h
FILE_END . . . . . . . . . . . .  Number  00000002h
FILE_READ_DATA . . . . . . . . .  Number  00000001h
FILE_SHARE_DELETE  . . . . . . .  Number  00000004h
FILE_SHARE_READ  . . . . . . . .  Number  00000001h
FILE_SHARE_WRITE . . . . . . . .  Number  00000002h
FILE_WRITE_DATA  . . . . . . . .  Number  00000002h
FOCUS_EVENT  . . . . . . . . . .  Number  00000010h
FORMAT_MESSAGE_ALLOCATE_BUFFER .  Number  00000100h
FORMAT_MESSAGE_FROM_SYSTEM . . .  Number  00001000h
FormatMessage  . . . . . . . . .  Text          FormatMessageA
GENERIC_ALL  . . . . . . . . . .  Number  10000000h
GENERIC_EXECUTE  . . . . . . . .  Number  20000000h
GENERIC_READ . . . . . . . . . .  Number  -80000000h
GENERIC_WRITE  . . . . . . . . .  Number  40000000h
GetCommandLine . . . . . . . . .  Text          GetCommandLineA
HANDLE . . . . . . . . . . . . .  Text          DWORD
HEAP_GENERATE_EXCEPTIONS . . . .  Number  00000004h
HEAP_GROWABLE  . . . . . . . . .  Number  00000002h
HEAP_NO_SERIALIZE  . . . . . . .  Number  00000001h
HEAP_REALLOC_IN_PLACE_ONLY . . .  Number  00000010h
HEAP_ZERO_MEMORY . . . . . . . .  Number  00000008h
IDABORT  . . . . . . . . . . . .  Number  00000003h
```

```
IDCANCEL . . . . . .  Number  00000002h
IDCLOSE  . . . . . .  Number  00000008h
IDCONTINUE . . . . . .  Number  0000000Bh
IDHELP . . . . . .  Number  00000009h
IDIGNORE . . . . . .  Number  00000005h
IDNO . . . . . .  Number  00000007h
IDOK . . . . . .  Number  00000001h
IDRETRY  . . . . . .  Number  00000004h
IDTIMEOUT  . . . . . .  Number  00007D00h
IDTRYAGAIN . . . . . .  Number  0000000Ah
IDYES  . . . . . .  Number  00000006h
INVALID_HANDLE_VALUE . . . . . .  Number  -00000001h
KBDOWN_FLAG  . . . . . .  Number  00000001h
KEY_EVENT  . . . . . .  Number  00000001h
KEY_MASKS  . . . . . .  Number  0000001Fh
LEFT_ALT_PRESSED . . . . . .  Number  00000002h
LEFT_CTRL_PRESSED  . . . . . .  Number  00000008h
MAX_BOUND  . . . . . .  Number  00000064h
MB_ABORTRETRYIGNORE  . . . . . .  Number  00000002h
MB_APPLMODAL . . . . . .  Number  00000000h
MB_CANCELTRYCONTINUE . . . . . .  Number  00000006h
MB_DEFBUTTON1  . . . . . .  Number  00000000h
MB_DEFBUTTON2  . . . . . .  Number  00000100h
MB_DEFBUTTON3  . . . . . .  Number  00000200h
MB_DEFBUTTON4  . . . . . .  Number  00000300h
MB_HELP  . . . . . .  Number  00004000h
MB_ICONASTERISK  . . . . . .  Number  00000040h
MB_ICONERROR . . . . . .  Number  00000010h
MB_ICONEXCLAMATION . . . . . .  Number  00000030h
MB_ICONHAND  . . . . . .  Number  00000010h
MB_ICONINFORMATION . . . . . .  Number  00000040h
MB_ICONQUESTION  . . . . . .  Number  00000020h
MB_ICONSTOP  . . . . . .  Number  00000010h
MB_ICONWARNING . . . . . .  Number  00000030h
MB_OKCANCEL  . . . . . .  Number  00000001h
MB_OK  . . . . . .  Number  00000000h
MB_RETRYCANCEL . . . . . .  Number  00000005h
MB_SYSTEMMODAL . . . . . .  Number  00001000h
MB_TASKMODAL . . . . . .  Number  00002000h
MB_USERICON  . . . . . .  Number  00000080h
MB_YESNOCANCEL . . . . . .  Number  00000003h
MB_YESNO . . . . . .  Number  00000004h
MENU_EVENT . . . . . .  Number  00000008h
MIN_A  . . . . . .  Number  0000005Ah
MIN_BOUND  . . . . . .  Number  00000000h
MIN_B  . . . . . .  Number  00000050h
MIN_C  . . . . . .  Number  00000046h
MIN_D  . . . . . .  Number  0000003Ch
```

```
MIN_F  . . . . . .  Number  00000032h
MOUSE_EVENT  . . . . . .  Number  00000002h
MessageBox . . . . . .  Text          MessageBoxA
NULL . . . . . .  Number  00000000h
NUMLOCK_ON . . . . . .  Number  00000020h
OPEN_ALWAYS  . . . . . .  Number  00000004h
OPEN_EXISTING  . . . . . .  Number  00000003h
PeekConsoleInput . . . . . .  Text          PeekConsoleInputA
RIGHT_ALT_PRESSED  . . . . . .  Number  00000001h
RIGHT_CTRL_PRESSED . . . . . .  Number  00000004h
ReadConsoleInput . . . . . .  Text          ReadConsoleInputA
ReadConsole  . . . . . .  Text          ReadConsoleA
SCROLLLOCK_ON  . . . . . .  Number  00000040h
SHIFT_MASK . . . . . .  Number  00000010h
SHIFT_PRESSED  . . . . . .  Number  00000010h
STD_ERROR_HANDLE . . . . . .  Number  -0000000Ch
STD_INPUT_HANDLE . . . . . .  Number  -0000000Ah
STD_OUTPUT_HANDLE  . . . . . .  Number  -0000000Bh
SetConsoleTitle  . . . . . .  Text          SetConsoleTitleA
TAB  . . . . . .  Number  00000009h
TRUE . . . . . .  Number  00000001h
TRUNCATE_EXISTING  . . . . . .  Number  00000005h
VK_11  . . . . . .  Number  000000BDh
VK_12  . . . . . .  Number  000000BBh
VK_ADD . . . . . .  Number  0000006Bh
VK_BACK  . . . . . .  Number  00000008h
VK_CANCEL  . . . . . .  Number  00000003h
VK_CAPITAL . . . . . .  Number  00000014h
VK_CLEAR . . . . . .  Number  0000000Ch
VK_CONTROL . . . . . .  Number  00000011h
VK_DECIMAL . . . . . .  Number  0000006Eh
VK_DELETE  . . . . . .  Number  0000002Eh
VK_DIVIDE  . . . . . .  Number  0000006Fh
VK_DOWN  . . . . . .  Number  00000028h
VK_END . . . . . .  Number  00000023h
VK_ESCAPE  . . . . . .  Number  0000001Bh
VK_EXECUTE . . . . . .  Number  0000002Bh
VK_F10 . . . . . .  Number  00000079h
VK_F11 . . . . . .  Number  0000007Ah
VK_F12 . . . . . .  Number  0000007Bh
VK_F13 . . . . . .  Number  0000007Ch
VK_F14 . . . . . .  Number  0000007Dh
VK_F15 . . . . . .  Number  0000007Eh
VK_F16 . . . . . .  Number  0000007Fh
VK_F17 . . . . . .  Number  00000080h
VK_F18 . . . . . .  Number  00000081h
VK_F19 . . . . . .  Number  00000082h
VK_F1  . . . . . .  Number  00000070h
```

```
VK_F20 . . . . . .  Number  00000083h
VK_F21 . . . . . .  Number  00000084h
VK_F22 . . . . . .  Number  00000085h
VK_F23 . . . . . .  Number  00000086h
VK_F24 . . . . . .  Number  00000087h
VK_F2  . . . . . .  Number  00000071h
VK_F3  . . . . . .  Number  00000072h
VK_F4  . . . . . .  Number  00000073h
VK_F5  . . . . . .  Number  00000074h
VK_F6  . . . . . .  Number  00000075h
VK_F7  . . . . . .  Number  00000076h
VK_F8  . . . . . .  Number  00000077h
VK_F9  . . . . . .  Number  00000078h
VK_HELP  . . . . . .  Number  0000002Fh
VK_HOME  . . . . . .  Number  00000024h
VK_INSERT  . . . . . .  Number  0000002Dh
VK_LBUTTON . . . . . .  Number  00000001h
VK_LCONTROL  . . . . . .  Number  000000A2h
VK_LEFT  . . . . . .  Number  00000025h
VK_LMENU . . . . . .  Number  000000A4h
VK_LSHIFT  . . . . . .  Number  000000A0h
VK_MENU  . . . . . .  Number  00000012h
VK_MULTIPLY  . . . . . .  Number  0000006Ah
VK_NEXT  . . . . . .  Number  00000022h
VK_NUMLOCK . . . . . .  Number  00000090h
VK_NUMPAD0 . . . . . .  Number  00000060h
VK_NUMPAD1 . . . . . .  Number  00000061h
VK_NUMPAD2 . . . . . .  Number  00000062h
VK_NUMPAD3 . . . . . .  Number  00000063h
VK_NUMPAD4 . . . . . .  Number  00000064h
VK_NUMPAD5 . . . . . .  Number  00000065h
VK_NUMPAD6 . . . . . .  Number  00000066h
VK_NUMPAD7 . . . . . .  Number  00000067h
VK_NUMPAD8 . . . . . .  Number  00000068h
VK_NUMPAD9 . . . . . .  Number  00000069h
VK_PAUSE . . . . . .  Number  00000013h
VK_PRINT . . . . . .  Number  0000002Ah
VK_PRIOR . . . . . .  Number  00000021h
VK_RBUTTON . . . . . .  Number  00000002h
VK_RCONTROL  . . . . . .  Number  000000A3h
VK_RETURN  . . . . . .  Number  0000000Dh
VK_RIGHT . . . . . .  Number  00000027h
VK_RMENU . . . . . .  Number  000000A5h
VK_RSHIFT  . . . . . .  Number  000000A1h
VK_SCROLL  . . . . . .  Number  00000091h
VK_SEPARATER . . . . . .  Number  0000006Ch
VK_SHIFT . . . . . .  Number  00000010h
VK_SNAPSHOT  . . . . . .  Number  0000002Ch
```

```
VK_SPACE . . . . . .  Number  00000020h
VK_SUBTRACT  . . . . . .  Number  0000006Dh
VK_TAB . . . . . .  Number  00000009h
VK_UP  . . . . . .  Number  00000026h
WINDOW_BUFFER_SIZE_EVENT . . . .  Number  00000004h
WriteConsoleOutputCharacter  . .  Text          WriteConsoleOutputCharacterA
WriteConsole . . . . . .  Text          WriteConsoleA
black  . . . . . .  Number  00000000h
blue . . . . . .  Number  00000001h
brown  . . . . . .  Number  00000006h
cyan . . . . . .  Number  00000003h
exit . . . . . .  Text          INVOKE ExitProcess,0
gradeA . . . . . .  Byte    00000000 _DATA
gradeB . . . . . .  Byte    00000002 _DATA
gradeC . . . . . .  Byte    00000004 _DATA
gradeD . . . . . .  Byte    00000006 _DATA
gradeFinal . . . . . .  Byte    0000000A _DATA
gradeF . . . . . .  Byte    00000008 _DATA
gray . . . . . .  Number  00000008h
green  . . . . . .  Number  00000002h
lightBlue  . . . . . .  Number  00000009h
lightCyan  . . . . . .  Number  0000000Bh
lightGray  . . . . . .  Number  00000007h
lightGreen . . . . . .  Number  0000000Ah
lightMagenta . . . . . .  Number  0000000Dh
lightRed . . . . . .  Number  0000000Ch
magenta  . . . . . .  Number  00000005h
msg  . . . . . .  Byte    00000021 _DATA
outOfBounds  . . . . . .  Byte    0000000C _DATA
red  . . . . . .  Number  00000004h
value  . . . . . .  DWord   0000002E _DATA
white  . . . . . .  Number  0000000Fh
wsprintf . . . . . .  Text          wsprintfA
yellow . . . . . .  Number  0000000Eh

          0 Warnings
          0 Errors
```

Source Code

```
;;     Author:       Marco Martinez
;;     Filename:     gradeCalc.asm
;;     Version:      1.0
;;     Description:  Create a procedure named CalcGrade that receives an integer value between 0 and
;;     100, and returns a single capital letter in the AL register. Preserve all other register values
;;     between calls to the procedure. The letter returned by the procedure should be according to the
;;     following ranges: 90 - 100 = A; 80 - 89 = B; 70 - 79 = C, 60 - 69 = D, 0 - 59 = F
;;     Write a test program that generates 10 random integers between 50 and 100, inclusive. Each
```

```
;;     time an integer is generated, pass it to the CalcGrade procedure. You can test your program
;;     using a debugger, or if you prefer to use the book’s library, you can display each integer and
;;     its corresponding letter grade. (The Irvine32 library is required for this solution program
;;     because it uses the RandomRange procedure.)
;;     Date:         10/28
;;
;;     Program Change Log
;;     ==================
;;     Name          Date          Description
;;     Marco  10/28  Create baseline for gradeCalc.asm
;;

INCLUDE Irvine32.inc
MIN_A EQU 90
MIN_B EQU 80
MIN_C EQU 70
MIN_D EQU 60
MIN_F EQU 50
MIN_BOUND EQU 0
MAX_BOUND EQU 100
.data
       gradeA BYTE "A",0
       gradeB BYTE "B",0
       gradeC BYTE "C",0
       gradeD BYTE "D",0
       gradeF BYTE "F",0
       gradeFinal BYTE ?,0
       outOfBounds BYTE "Value out of bounds.",0
       msg BYTE " equates to ",0

.code
main PROC
       call Randomize
       mov ecx, 10
L1:
       call DetermineInt
       call Calc
       call DisplayGrade
       loop L1
       exit
main ENDP

;-----------------------------------------------------
DetermineInt PROC
;
; Determine the integers to be used for grade calculation (random)
; Receives: nothing
; Returns: nothing
```

```
;-----------------------------------------------------
       mov eax,50
     call RandomRange
       add eax,51
       mov edx,eax
       Call WriteDec
       ret
DetermineInt ENDP

;-----------------------------------------------------
DisplayGrade PROC
;
; Displays the grade for each integer
; Receives: nothing
; Returns: nothing
;-----------------------------------------------------
       mov edx,0
       mov edx,OFFSET msg
       call WriteString
       mov gradeFinal,al
       mov edx,OFFSET gradeFinal
       call WriteString
       call Crlf
       ret
DisplayGrade ENDP

;-----------------------------------------------------
Calc PROC
;
; Calculates which grade the integer represents
; Receives: nothing
; Returns: nothing
;-----------------------------------------------------
       cmp eax,MAX_BOUND
       ja OutBounds
       cmp eax,MIN_BOUND
       jb OutBounds
       cmp eax,MIN_A
       jae EarnedA
       cmp eax,MIN_B
       jae EarnedB
       cmp eax,MIN_C
       jae EarnedC
       cmp eax,MIN_D
       jae EarnedD
       cmp eax,MIN_F
       jae EarnedF
EarnedA:
```

```
        mov al,gradeA
        jmp Stop
EarnedB:
        mov al,gradeB
        jmp Stop
EarnedC:
        mov al,gradeC
        jmp Stop
EarnedD:
        mov al,gradeD
        jmp Stop
EarnedF:
        mov al,gradeF
        jmp Stop
OutBounds:
        mov edx,OFFSET outOfBounds
        call WriteString
        call Crlf
Stop:
        ret
Calc ENDP
END main
```

Screenshot