Marco Martinez
CISP 310
Assignment 3a

HIERARCHY CHART

3.0 DetermineLargest
	3.1 FindLargest(in arrayPtr as array of integers, in arraySize as integer)

PSEUDO CODE

Main Module

```
Begin
	DECLARE array1 as Array of Integers {0,10,20}
	DECLARE array2 as Array of Integers {-10,0,10}
	DECLARE array3 as Array of Integers {400,200,3000,-2000,40}
	DECLARE prompt1 as String "Of the set {"
	DECLARE prompt2 as String "} the highest number is "
	DECLARE comma as String ","

	Save reg
	Set edx as OFFSET prompt1
	Call WriteString
	Set esi as OFFSET array1
	Set ecx as 0
Do
	Set eax as [esi]
	Call WriteInt
	Set esi as esi + 4
	Set edx as OFFSET comma
	Set ecx as ecx + 1
	If (ecx != LENGTHOF array1)
		call writeString
	Endif
	While (ecx < LENGTHOF array1)
End DoWhile

	Call FindLargest(in OFFSET array1,in LENGTHOF array1)
	Set edx as OFFSET prompt2
	Call WriteString
	Call WriteInt
	Call Crlf
	Load reg

	Save reg
	Set edx as OFFSET prompt1
	Call WriteString
	Set esi as OFFSET array2
	Set ecx as 0
Do
	Set eax as [esi]
	Call WriteInt
	Set esi as esi + 4
	Set edx as OFFSET comma
	Set ecx as ecx + 1
	If (ecx != LENGTHOF array2)
		call writeString
	Endif
	While (ecx < LENGTHOF array2)
End DoWhile

	Call FindLargest(in OFFSET array2,in LENGTHOF array2)
	Set edx as OFFSET prompt2
	Call WriteString
	Call WriteInt
	Call Crlf
	Load reg
	Save reg
	Set edx as OFFSET prompt1
```

```
        Call WriteString
        Set esi as OFFSET array3
        Set ecx as 0
Do
        Set eax as [esi]
        Call WriteInt
        Set esi as esi + 4
        Set edx as OFFSET comma
        Set ecx as ecx + 1
        If (ecx != LENGTHOF array3)
                call writeString
        Endif
        While (ecx < LENGTHOF array3)
End DoWhile

        Call FindLargest(in OFFSET array3,in LENGTHOF array3)
        Set edx as OFFSET prompt2
        Call WriteString
        Call WriteInt
        Call Crlf
        Load reg
End Main

FindLargest Module (in arrayPtr as array of integers, in arraySize as integer)
Begin
        Set esi as arrayPtr
        Set eax as [esi]
        Set ecx as arraySize
        If ecx < 1
                Return
        End If

        While (ecx > 0)
                If (eax >= [esi])
                        Set eax as [esi]
                End If
        Set esi as esi + 4
        Set ecx as ecx – 1
        End While
Return
```

ASSEMBLY SOURCE CODE

```
;;      Author:              Marco Martinez
;;      Filename:            determineLargest.asm
;;      Version:             1.0
;;      Description:  Create a procedure named FindLargest that receives two parameters: a pointer to a
signed
;;                           doubleword array, and a count of the array’s length. The procedure must
return the value of
;;                           the largest array member in EAX. Use the PROC directive with a parameter
list when declaring
;;                           the procedure. Preserve all registers (except EAX) that are modified by the
procedure.
;;                           Write a test program that calls FindLargest and passes three different
arrays of different
;;                           lengths. Be sure to include negative values in your arrays. Create a PROTO
declaration for
;;                           FindLargest.
;;      Date:         12/2
;;
;;      Program Change Log
;;      ==================
;;      Name          Date          Description
;;      Marco  12/2          Create baseline for findLargest.asm
;;
```

```
INCLUDE Irvine32.inc

.data
array1 DWORD 0,10,20
array2 DWORD -10,0,10
array3 DWORD 400,200,3000,-2000,40
prompt1 BYTE "Of the set {",0
prompt2 BYTE "} the highest number is ",0
comma BYTE ",",0

.code
FindLargest PROTO,arrayPtr:PTR DWORD,arraySize:DWORD

main PROC

	popad
	mov edx,OFFSET prompt1
	call WriteString
	mov esi,OFFSET array1
	mov ecx,0
L1:
	mov eax,[esi]
	call WriteInt
	add esi,4
	mov edx,OFFSET comma
	inc ecx
	cmp ecx,LENGTHOF array1
	je Skip1
	call writeString
Skip1:
	cmp ecx,LENGTHOF array1
	jl L1
	invoke FindLargest,OFFSET array1,LENGTHOF array1
	mov edx,OFFSET prompt2
	call WriteString
	call WriteInt
	call Crlf
	pushad

	popad
	mov edx,OFFSET prompt1
	call WriteString
	mov esi,OFFSET array2
	mov ecx,0
L2:
	mov eax,[esi]
	call WriteInt
	add esi,4
	mov edx,OFFSET comma
	inc ecx
	cmp ecx,LENGTHOF array2
	je Skip2
	call writeString
Skip2:
	cmp ecx,LENGTHOF array2
	jl L2
	invoke FindLargest,OFFSET array2,LENGTHOF array2
	mov edx,OFFSET prompt2
	call WriteString
	call WriteInt
	call Crlf
	pushad
```

```
	popad
	mov edx,OFFSET prompt1
	call WriteString
	mov esi,OFFSET array3
	mov ecx,0
L3:
	mov eax,[esi]
	call WriteInt
	add esi,4
	mov edx,OFFSET comma
	inc ecx
	cmp ecx,LENGTHOF array3
	je Skip3
	call writeString
Skip3:
	cmp ecx,LENGTHOF array3
	jl L3
	invoke FindLargest,OFFSET array3,LENGTHOF array3
	mov edx,OFFSET prompt2
	call WriteString
	call WriteInt
	call Crlf
	pushad

	exit
main ENDP

FindLargest PROC USES ecx esi,
	arrayPtr:PTR DWORD,
	arraySize:DWORD
	mov esi,arrayPtr
	mov eax,[esi]
	mov ecx,arraySize
	cmp ecx,1
	je Conclude
L1:
	cmp eax,[esi]
	jge GreaterOrEquals
	mov eax,[esi]
GreaterOrEquals:
	add esi,4
	loop L1
Conclude:
	ret
FindLargest ENDP
END main
```

ASSEMBLY LISTING CODE

```
Microsoft (R) Macro Assembler Version 14.15.26732.1           12/11/18 16:02:26
..\..\..\..\Documents\School Work\P310\thirdProject\determineLargest.asm  Page 1 - 1


                              ;;     Author:        Marco Martinez
                              ;;     Filename:              determineLargest.asm
                              ;;     Version:               1.0
                              ;;     Description:   Create a procedure named FindLargest that receives two
parameters: a pointer to a signed
                              ;;                            doubleword array, and a count of the array's
length. The procedure must return the value of
                              ;;                            the largest array member in EAX. Use the PROC
directive with a parameter list when declaring
                              ;;                            the procedure. Preserve all registers (except EAX)
that are modified by the procedure.
                              ;;                            Write a test program that calls FindLargest and
passes three different arrays of different
                              ;;                            lengths. Be sure to include negative values in
your arrays. Create a PROTO declaration for
                              ;;                            FindLargest.
                              ;;     Date:          12/2
                              ;;
                              ;;     Program Change Log
                              ;;     ==================
                              ;;     Name           Date           Description
                              ;;     Marco  12/2           Create baseline for findLargest.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
 00000000 00000000           array1 DWORD 0,10,20
          0000000A
          00000014
 0000000C FFFFFFF6           array2 DWORD -10,0,10
          00000000
          0000000A
 00000018 00000190           array3 DWORD 400,200,3000,-2000,40
          000000C8
          00000BB8
          FFFFF830
          00000028
 0000002C 4F 66 20 74 68     prompt1 BYTE "Of the set {",0
          65 20 73 65 74
          20 7B 00
 00000039 7D 20 74 68 65     prompt2 BYTE "} the highest number is ",0
          20 68 69 67 68
          65 73 74 20 6E
          75 6D 62 65 72
```

```
         20 69 73 20 00
00000052 2C 00                    comma BYTE ",",0

00000000                    .code
                            FindLargest PROTO,arrayPtr:PTR DWORD,arraySize:DWORD

00000000                    main PROC

00000000  60                      pushad
00000001  BA 0000002C R           mov edx,OFFSET prompt1
00000006  E8 00000000 E           call WriteString
0000000B  BE 00000000 R           mov esi,OFFSET array1
00000010  B9 00000000                     mov ecx,0
00000015                    L1:
00000015  8B 06                   mov eax,[esi]
00000017  E8 00000000 E           call WriteInt
0000001C  83 C6 04                add esi,4
0000001F  BA 00000052 R           mov edx,OFFSET comma
00000024  41                      inc ecx
00000025  83 F9 03                cmp ecx,LENGTHOF array1
00000028  74 05                   je Skip1
0000002A  E8 00000000 E           call writeString
0000002F                    Skip1:
0000002F  83 F9 03                cmp ecx,LENGTHOF array1
00000032  7C E1                   jl L1
                                  invoke FindLargest,OFFSET array1,LENGTHOF array1
00000034  6A 03         *       push   +000000003h
00000036  68 00000000 R   *     push   dword  ptr OFFSET FLAT: array1
0000003B  E8 000000C6           *       call   FindLargest
00000040  BA 00000039 R           mov edx,OFFSET prompt2
00000045  E8 00000000 E           call WriteString
0000004A  E8 00000000 E           call WriteInt
0000004F  E8 00000000 E           call Crlf
00000054  61                      popad

00000055  60                      pushad
00000056  BA 0000002C R           mov edx,OFFSET prompt1
0000005B  E8 00000000 E           call WriteString
00000060  BE 0000000C R           mov esi,OFFSET array2
00000065  B9 00000000                     mov ecx,0
0000006A                    L2:
0000006A  8B 06                   mov eax,[esi]
0000006C  E8 00000000 E           call WriteInt
00000071  83 C6 04                add esi,4
00000074  BA 00000052 R           mov edx,OFFSET comma
00000079  41                      inc ecx
0000007A  83 F9 03                cmp ecx,LENGTHOF array2
0000007D  74 05                   je Skip2
0000007F  E8 00000000 E           call writeString
00000084                    Skip2:
00000084  83 F9 03                cmp ecx,LENGTHOF array2
00000087  7C E1                   jl L2
                                  invoke FindLargest,OFFSET array2,LENGTHOF array2
00000089  6A 03         *       push   +000000003h
0000008B  68 0000000C R   *     push   dword  ptr OFFSET FLAT: array2
00000090  E8 00000071           *       call   FindLargest
00000095  BA 00000039 R           mov edx,OFFSET prompt2
0000009A  E8 00000000 E           call WriteString
0000009F  E8 00000000 E           call WriteInt
000000A4  E8 00000000 E           call Crlf
000000A9  61                      popad

000000AA  60                      pushad
000000AB  BA 0000002C R           mov edx,OFFSET prompt1
000000B0  E8 00000000 E           call WriteString
000000B5  BE 00000018 R           mov esi,OFFSET array3
000000BA  B9 00000000                     mov ecx,0
000000BF                    L3:
```

```
000000BF  8B 06                     mov eax,[esi]
000000C1  E8 00000000 E             call WriteInt
000000C6  83 C6 04                  add esi,4
000000C9  BA 00000052 R             mov edx,OFFSET comma
000000CE  41                        inc ecx
000000CF  83 F9 05                  cmp ecx,LENGTHOF array3
000000D2  74 05                     je Skip3
000000D4  E8 00000000 E             call writeString
000000D9                    Skip3:
000000D9  83 F9 05                  cmp ecx,LENGTHOF array3
000000DC  7C E1                     jl L3
                                    invoke FindLargest,OFFSET array3,LENGTHOF array3
000000DE  6A 05          *        push   +000000005h
000000E0  68 00000018 R   *      push   dword  ptr OFFSET FLAT: array3
000000E5  E8 0000001C           *       call   FindLargest
000000EA  BA 00000039 R             mov edx,OFFSET prompt2
000000EF  E8 00000000 E             call WriteString
000000F4  E8 00000000 E             call WriteInt
000000F9  E8 00000000 E             call Crlf
000000FE  61                        popad

                                    exit
000000FF  6A 00          *        push   +000000000h
00000101  E8 00000000 E   *      call   ExitProcess
00000106                    main ENDP

00000106                    FindLargest PROC USES ecx esi,
                                    arrayPtr:PTR DWORD,
                                    arraySize:DWORD
00000106  55             *       push   ebp
00000107  8B EC          *       mov    ebp, esp
00000109  51             *       push   ecx
0000010A  56             *       push   esi
0000010B  8B 75 08                  mov esi,arrayPtr
0000010E  8B 06                     mov eax,[esi]
00000110  8B 4D 0C                  mov ecx,arraySize
00000113  83 F9 01                  cmp ecx,1
00000116  74 0B                     je Conclude
00000118                    L1:
00000118  3B 06                     cmp eax,[esi]
0000011A  7D 02                     jge GreaterOrEquals
0000011C  8B 06                     mov eax,[esi]
0000011E                    GreaterOrEquals:
0000011E  83 C6 04                  add esi,4
00000121  E2 F5                     loop L1
00000123                    Conclude:
                                    ret
00000123  5E             *       pop    esi
00000124  59             *       pop    ecx
00000125  C9             *       leave
00000126  C2 0008        *       ret    00008h
00000129                    FindLargest 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  00000054 Para   Public  'DATA'
_TEXT  . . . . . . . . . . . . .    32 Bit  00000129 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
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
FindLargest  . . . . . . . . . .    P Near  00000106 _TEXT       Length= 00000023 Public STDCALL
  arrayPtr . . . . . . . . . . .    DWord   bp + 00000008
  arraySize  . . . . . . . . . .    DWord   bp + 0000000C
  L1 . . . . . . . . . . . . . .    L Near  00000118 _TEXT
  GreaterOrEquals  . . . . . . .    L Near  0000011E _TEXT
  Conclude . . . . . . . . . . .    L Near  00000123 _TEXT
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
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= 00000106 Public STDCALL
  L1 . . . . . .    L Near   00000015 _TEXT
  Skip1  . . . . . .    L Near   0000002F _TEXT
  L2 . . . . . .    L Near   0000006A _TEXT
  Skip2  . . . . . .    L Near   00000084 _TEXT
  L3 . . . . . .    L Near   000000BF _TEXT
  Skip3  . . . . . .    L Near   000000D9 _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
```

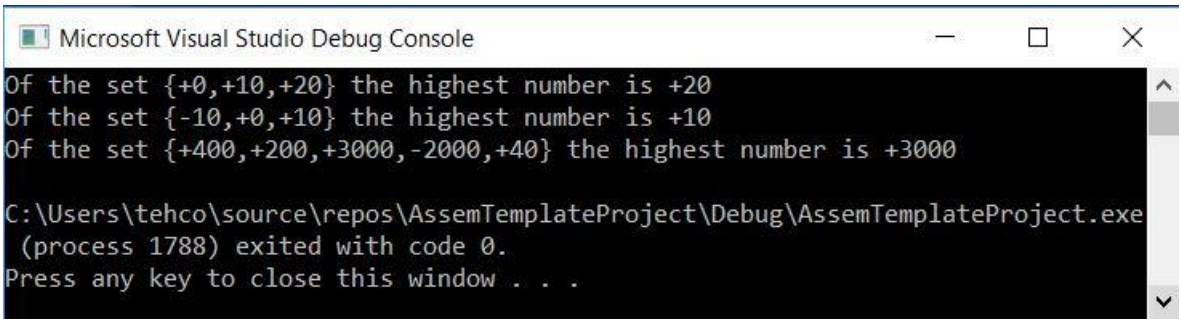
```
FILE_WRITE_DATA  . . . . . .    Number  00000002h
FOCUS_EVENT  . . . . . .    Number  00000010h
FORMAT_MESSAGE_ALLOCATE_BUFFER .    Number  00000100h
FORMAT_MESSAGE_FROM_SYSTEM . . .    Number  00001000h
FormatMessage  . . . . . .    Text    FormatMessageA
GENERIC_ALL  . . . . . .    Number  10000000h
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
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
array1 . . . . . .        DWord    00000000 _DATA
array2 . . . . . .        DWord    0000000C _DATA
array3 . . . . . .        DWord    00000018 _DATA
black  . . . . . .        Number   00000000h
blue . . . . . .          Number   00000001h
brown  . . . . . .        Number   00000006h
comma  . . . . . .        Byte     00000052 _DATA
cyan . . . . . .          Number   00000003h
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
prompt1  . . . . . .      Byte     0000002C _DATA
prompt2  . . . . . .      Byte     00000039 _DATA
red  . . . . . .          Number   00000004h
white  . . . . . .        Number   0000000Fh
wsprintf . . . . . .      Text     wsprintfA
yellow . . . . . .        Number   0000000Eh

          0 Warnings
          0 Errors
```

CONSOLE SCREENSHOT



Hierarchy Chart

3.0 Counting
	3.1 CountNearMatches(in arrayPtr1,arrayPtr2 as Array of Integers, in arraySize as Integer, in margin as Integer)

Pseudocode

Main Module

```
Begin
	DECLARE array1 as Array of Integers {-10,10,20}
	DECLARE array2 as Array of Integers {-14,12,0}
	DECLARE array3 as Array of Integers {-100,0,100,200}
	DECLARE array4 as Array of Integers {-100,10,100,200}
	DECLARE diff as Integer 4
	DECLARE msg1 as String "The margin is plus or minus "
	DECLARE msg2 as String "The arrays {"
	DECLARE msg3 as String "} and {"
	DECLARE msg4 as String "} contain "
	DECLARE msg5 as String " near matches."
	DECLARE comma as String ","
	DECLARE period as String "."

	Save reg
	Set edx as OFFSET msg1
	Call WriteString
	Set eax as diff
	Call writeInt
	Set edx as OFFSET period
	Call writeString
	Call crlf
	Set edx as OFFSET msg2
	Call WriteString
	Set esi as OFFSET array1
	Set ecx as 0
Do
	Set eax as [esi]
	Call WriteInt
	Set esi as esi + 4
	Set edx as OFFSET comma
	Set ecx as ecx + 1
	If (ecx != LENGTHOF array1)
		Call writeString
	EndIf
	While (ecx < LENGTHOF array1)

	Set edx as OFFSET msg3
	Call WriteString
	Set esi as OFFSET array2
	Set ecx as 0
Do
	Set eax,[esi]
	Call WriteInt
	Set esi as esi + 4
	Set edx as OFFSET comma
	Set ecx as ecx + 1
	If (ecx != LENGTHOF array2)
		call writeString
	Endif
	While (ecx < LENGTHOF array2)
	mov edx,OFFSET msg4
	call writeString
	invoke CountNearMatches, OFFSET array1, OFFSET array2, LENGTHOF array1, diff
	call writeInt
	mov edx,OFFSET msg5
	call writeString
	call crlf
	Load reg
```

```
        Save reg
        Set edx as OFFSET msg1
        Call WriteString
        Set eax as diff
        Call writeInt
        Set edx as OFFSET period
        Call writeString
        Call crlf
        Set edx as OFFSET msg2
        Call WriteString
        Set esi as OFFSET array3
        Set ecx as 0
Do
        Set eax as [esi]
        Call WriteInt
        Set esi as esi + 4
        Set edx as OFFSET comma
        Set ecx as ecx + 1
        If (ecx != LENGTHOF array3)
                Call writeString
        EndIf
        While (ecx < LENGTHOF array3)

        Set edx as OFFSET msg3
        Call WriteString
        Set esi as OFFSET array4
        Set ecx as 0
Do
        Set eax,[esi]
        Call WriteInt
        Set esi as esi + 4
        Set edx as OFFSET comma
        Set ecx as ecx + 1
        If (ecx != LENGTHOF array4)
                call writeString
        Endif
        While (ecx < LENGTHOF array4)
        mov edx,OFFSET msg4
        call writeString
        invoke CountNearMatches, OFFSET array3, OFFSET array4, LENGTHOF array3, diff
        call writeInt
        mov edx,OFFSET msg5
        call writeString
        call crlf
        Load reg

End main

CountNearMatches Module (in arrayPtr1,arrayPtr2 as array of integers, in arraySize as integer, in margin as
integer)
Begin
        Declare count as Local Integer

        Set count as 0
        Set esi as arrayPtr1
        Set edi as arrayPtr2
        Set ecx as arraySize

        While (ecx > 0)
                Set eax as [esi]
                Set ebx as [edi]


                If (eax < ebx)
                        Set edx as eax
                        Set eax as ebx
                        Set ebx as edx
                End If
```

```
                Set eax as eax - ebx
                Set edx as margin

                If (eax <= edx)
                        Set eax as 1
                        Set count as count + eax
                End If

                Set esi as esi + 4
                Set edi as edi + 4
        End While
        Set eax as count
Return
```

ASSEMBLY SOURCE CODE

```
;;       Author:            Marco Martinez
;;       Filename:          counting.asm
;;       Version:           1.0
;;       Description:       Write a procedure named CountNearMatches that receives pointers to two arrays of signed doublewords,
;;                          a parameter that indicates the length of the two arrays, and a parameter that indicates the
;;                          maximum allowed difference (called diff) between any two matching elements. For each element x
;;                          in the first array, if the difference between it and the corresponding y in the second array is less
;;                          than or equal to diff, increment a count. At the end, return a count of the number of nearly matching
;;                          array elements in EAX. Write a test program that calls CountNearMatches and passes pointers
;;                          to two different pairs of arrays. Use the INVOKE statement to call your procedure and pass stack
;;                          parameters. Create a PROTO declaration for CountMatches. Save and restore any registers (other
;;                          than EAX) changed by your procedure.
;;       Date:              12/2
;;
;;       Program Change Log
;;       ==================
;;       Name               Date              Description
;;       Marco              12/2              Create baseline for counting.asm
;;

INCLUDE Irvine32.inc

.data
array1 DWORD -10,10,20
array2 DWORD -14,12,0
array3 DWORD -100,0,100,200
array4 DWORD -100,10,100,200
diff DWORD 4
msg1 BYTE "The margin is plus or minus ",0
msg2 BYTE "The arrays {",0
msg3 BYTE "} and {",0
msg4 BYTE "} contain ",0
msg5 BYTE " near matches.",0
comma BYTE ",",0
period BYTE ".",0

.code
CountNearMatches PROTO,
         arrayPtr1:PTR DWORD,
         arrayPtr2:PTR DWORD,
         arraySize:DWORD,
         margin:DWORD

main PROC
         popad
         mov edx,OFFSET msg1
         call WriteString
         mov eax,diff
         call writeInt
         mov edx,OFFSET period
         call writeString
         call crlf
         mov edx,OFFSET msg2
         call WriteString
         mov esi,OFFSET array1
         mov ecx,0
L1:
         mov eax,[esi]
         call WriteInt
         add esi,4
         mov edx,OFFSET comma
         inc ecx
         cmp ecx,LENGTHOF array1
         je Skip1
         call writeString
Skip1:
```

```
          cmp ecx,LENGTHOF array1
          jl L1
          mov edx,OFFSET msg3
          call WriteString
          mov esi,OFFSET array2
          mov ecx,0
L2:
          mov eax,[esi]
          call WriteInt
          add esi,4
          mov edx,OFFSET comma
          inc ecx
          cmp ecx,LENGTHOF array2
          je Skip2
          call writeString
Skip2:
          cmp ecx,LENGTHOF array2
          jl L2
          mov edx,OFFSET msg4
          call writeString
          invoke CountNearMatches, OFFSET array1, OFFSET array2, LENGTHOF array1, diff
          call writeInt
          mov edx,OFFSET msg5
          call writeString
          call crlf
          pushad

          popad
          mov edx,OFFSET msg2
          call WriteString
          mov esi,OFFSET array3
          mov ecx,0
L3:
          mov eax,[esi]
          call WriteInt
          add esi,4
          mov edx,OFFSET comma
          inc ecx
          cmp ecx,LENGTHOF array3
          je Skip3
          call writeString
Skip3:
          cmp ecx,LENGTHOF array3
          jl L3
          mov edx,OFFSET msg3
          call WriteString
          mov esi,OFFSET array4
          mov ecx,0
L4:
          mov eax,[esi]
          call WriteInt
          add esi,4
          mov edx,OFFSET comma
          inc ecx
          cmp ecx,LENGTHOF array4
          je Skip4
          call writeString
Skip4:
          cmp ecx,LENGTHOF array4
          jl L4
          mov edx,OFFSET msg4
          call writeString
          invoke CountNearMatches, OFFSET array3, OFFSET array4, LENGTHOF array3, diff
          call writeInt
          mov edx,OFFSET msg5
          call writeString
          call crlf
          pushad

          exit
main ENDP

CountNearMatches PROC USES esi edi ebx ecx edx,
          arrayPtr1:PTR DWORD,
          arrayPtr2:PTR DWORD,
          arraySize:DWORD,
          margin:DWORD
          LOCAL count:DWORD

          mov count,0
          mov esi,arrayPtr1
          mov edi,arrayPtr2
          mov ecx,arraySize
L1:
          mov eax,[esi]
          mov ebx,[edi]
          cmp eax,ebx
```

```
        jg Subtraction
        mov edx,eax
        mov eax,ebx
        mov ebx,edx
Subtraction:
        sub eax,ebx
        mov edx,margin
        cmp eax,edx
        jg Reset
        mov eax,1
        add count,eax
Reset:
        add esi,4
        add edi,4
        loop L1
        mov eax,count
        ret
CountNearMatches ENDP
END main
```

ASSEMBLY LISTING FILE

```
Microsoft (R) Macro Assembler Version 14.15.26732.1           12/08/18 20:36:05
..\..\..\..\Documents\School Work\P310\thirdProject\Counting.asm  Page 1 - 1


                              ;;     Author:       Marco Martinez
                              ;;     Filename:             counting.asm
                              ;;     Version:              1.0
                              ;;     Description:  Write a procedure named CountNearMatches that receives
pointers to two arrays of signed doublewords,
                              ;;                           a parameter that indicates the length of the two
arrays, and a parameter that indicates the
                              ;;                           maximum allowed difference (called diff) between
any two matching elements. For each element x
                              ;;                           in the first array, if the difference between it
and the corresponding y in the second array is less
                              ;;                           than or equal to diff, increment a count. At the
end, return a count of the number of nearly matching
                              ;;                           array elements in EAX. Write a test program that
calls CountNearMatches and passes pointers
                              ;;                           to two different pairs of arrays. Use the INVOKE
statement to call your procedure and pass stack
                              ;;                           parameters. Create a PROTO declaration for
CountMatches. Save and restore any registers (other
                              ;;                           than EAX) changed by your procedure.
                              ;;     Date:         12/2
                              ;;
                              ;;     Program Change Log
                              ;;     ==================
                              ;;     Name          Date          Description
                              ;;     Marco  12/2          Create baseline for counting.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
00000000 FFFFFFF6           array1 DWORD -10,10,20
         0000000A
         00000014
0000000C FFFFFFF2           array2 DWORD -14,12,0
         0000000C
         00000000
00000018 00000004           diff DWORD 4
0000001C 4E 75 6D 62 65     message BYTE "Number of near matches between {-10,10,20} and {-14,12,0}: ",0
         72 20 6F 66 20
         6E 65 61 72 20
         6D 61 74 63 68
         65 73 20 62 65
         74 77 65 65 6E
         20 7B 2D 31 30
         2C 31 30 2C 32
         30 7D 20 61 6E
         64 20 7B 2D 31
         34 2C 31 32 2C
         30 7D 3A 20 00

00000000                    .code
                            CountNearMatches PROTO,
                                   arrayPtr1:PTR DWORD,
                                   arrayPtr2:PTR DWORD,
                                   arraySize:DWORD,
                                   margin:DWORD

00000000                    main PROC

                                   invoke CountNearMatches, OFFSET array1, OFFSET array2, LENGTHOF array1,
diff
00000000  FF 35 00000018 R *    push   diff
00000006  6A 03         *       push   +000000003h
00000008  68 0000000C R   *     push   dword  ptr OFFSET FLAT: array2
0000000D  68 00000000 R   *     push   dword  ptr OFFSET FLAT: array1
00000012  E8 00000020          *       call   CountNearMatches
00000017  BA 0000001C R            mov edx, OFFSET message
0000001C  E8 00000000 E            call WriteString
00000021  E8 00000000 E            call WriteInt
00000026  E8 00000000 E            call Crlf
0000002B  E8 00000000 E            call Crlf


                                   exit
00000030  6A 00         *       push   +000000000h
00000032  E8 00000000 E   *     call   ExitProcess
00000037                    main ENDP

00000037                    CountNearMatches PROC USES esi edi ebx ecx edx,
                                   arrayPtr1:PTR DWORD,
                                   arrayPtr2:PTR DWORD,
                                   arraySize:DWORD,
                                   margin:DWORD
                                   LOCAL count:DWORD

00000037  55            *       push   ebp
00000038  8B EC         *       mov    ebp, esp
0000003A  83 C4 FC      *       add    esp, 0FFFFFFFCh
0000003D  56            *       push   esi
0000003E  57            *       push   edi
0000003F  53            *       push   ebx
00000040  51            *       push   ecx
00000041  52            *       push   edx
00000042  8B 75 08                 mov esi,arrayPtr1
00000045  8B 7D 0C                 mov edi,arrayPtr2
00000048  8B 4D 10                 mov ecx,arraySize
0000004B                    L1:
```

```
0000004B  8B 06                       mov eax,[esi]
0000004D  8B 1F                       mov ebx,[edi]
0000004F  3B C3                       cmp eax,ebx
00000051  7F 06                       jg Subtraction
00000053  8B D0                       mov edx,eax
00000055  8B C3                       mov eax,ebx
00000057  8B DA                       mov ebx,edx
00000059                          Subtraction:
00000059  2B C3                       sub eax,ebx
0000005B  8B 55 14                    mov edx,margin
0000005E  3B C2                       cmp eax,edx
00000060  7F 08                       jg Reset
00000062  B8 00000001                         mov eax,1
00000067  01 45 FC                    add count,eax
0000006A                          Reset:
0000006A  83 C6 04                    add esi,4
0000006D  83 C7 04                    add edi,4
00000070  E2 D9                       loop L1
00000072  8B 45 FC                    mov eax,count
                                      ret
00000075  5A            *          pop    edx
00000076  59            *          pop    ecx
00000077  5B            *          pop    ebx
00000078  5F            *          pop    edi
00000079  5E            *          pop    esi
0000007A  C9            *          leave
0000007B  C2 0010       *          ret    00010h
0000007E                          CountNearMatches 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  00000058 Para   Public  'DATA'
_TEXT  . . . . . . . . . . . . .    32 Bit  0000007E 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
CountNearMatches . . . . . . . .    P Near  00000037 _TEXT       Length= 00000047 Public STDCALL
  arrayPtr1  . . . . . . . . . .    DWord   bp + 00000008
  arrayPtr2  . . . . . . . . . .    DWord   bp + 0000000C
  arraySize  . . . . . . . . . .    DWord   bp + 00000010
  margin . . . . . . . . . . . .    DWord   bp + 00000014
  count  . . . . . . . . . . . .    DWord   bp - 00000004
  L1 . . . . . . . . . . . . . .    L Near  0000004B _TEXT
  Subtraction  . . . . . . . . .    L Near  00000059 _TEXT
  Reset  . . . . . . . . . . . .    L Near  0000006A _TEXT
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
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
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= 00000037 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
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
array1 . . . . . .        DWord    00000000 _DATA
array2 . . . . . .        DWord    0000000C _DATA
black  . . . . . .        Number   00000000h
blue . . . . . .          Number   00000001h
brown  . . . . . .        Number   00000006h
cyan . . . . . .          Number   00000003h
diff . . . . . .          DWord    00000018 _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
message  . . . . . .      Byte     0000001C _DATA
red  . . . . . .          Number   00000004h
white  . . . . . .        Number   0000000Fh
wsprintf . . . . . .      Text     wsprintfA
yellow . . . . . .        Number   0000000Eh

          0 Warnings
          0 Errors
```

CONSOLE SCREENSHOT