Pseudo Code

```
Main Module
Begin
       INCLUDE Irvine32.inc
       Declare fib as array[1...7] of integers
       Declare counter as integer = 0
       Declare fibStart as String = "fib("
       Declare fibEnd as String = ")="
       Declare space as String = " "
       Save eax
       Set eax as 10
       Do
              Save reg
              Set ebx as 0
              Set edx as 1
              Set esi as OFFSET fib
              Set [esi],edx
              Set esi as esi + 4
              Set ecx as 6
              Do
                     Set eax as ebx
                     Set eax as eax + edx
                     Set [esi] as eax
                     Set esi as esi + 4
                     Set ebx as edx
                     Set edx as eax
              While(ecx > 0)
              Load reg
              Save reg
              Set ebx as 0
              Set esi as OFFSET fib
              Set edx as OFFSET fibStart
              Call writeString
              Set eax as ebx
              Set eax as eax + 1
              Call writeInt
              Set edx as OFFSET fibEnd
              Call writeString
              Set eax as [esi]
              Call writeInt
              Set edx as OFFSET space
              Call writeString
              Set ebx as ebx + 1
              Set ecx as 6
              Do
                     Set edx as OFFSET fibStart
                     Call writeString
                     Set eax as ebx
                     Set eax as eax + 1
                     Call writeInt
                     Set edx as OFFSET fibEnd
                     Call writeString
                     Set eax as [esi+ebx*4]
                     Call writeInt
```

```
Set edx as OFFSET space
                      Call writeString
                      Set ebx as ebx + 1
                      Set ecx as ecx - 1
               While(ecx<0)
               Load Reg
               Call crlf
               Set eax as eax - 1
       While (eax < 0)
       Load eax
End Main
Listing File
Microsoft (R) Macro Assembler Version 14.15.26732.1
                                                       12/06/18 09:59:26
..\..\Documents\School Work\P310\secondProject\ASM 2b\FibonacciNumbers.asm Page 1 - 1
                                      Author: Marco Martinez
                              ;;
                                      Program:
                                                     FibonacciNumbers.asm
                              ;;
                                      Date: 10/14/2018
                                      Purpose:
                                                     Write a program that uses a loop to calculate the first seven
values of the Fibonacci number sequence,
                                                     described by the following formula: Fib(1) = 1, Fib(2) = 1, Fib(n)
                              ;;
= Fib(n-1) + Fib(n-2).
                              ;;
                                      Software Change Record
                              ;;
                                      Name
                                                     Date
                                                                    What
                              ;;
                                      Marco 10/14 Baseline for FibonacciNumbers.asm
                                      Marco 12/3
                                                            Adjust for reporting
                              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
                          С
00000000
                              .data
```

fib DWORD 7 DUP(0)

00000000 00000007 [

00000000

```
0000001C 00000000
                                    counter DWORD 0
00000020 66 69 62 28 00
                                    fibStart BYTE "fib(",0
00000025 29 3D 00
                                    fibEnd BYTE ")=",0
                                           space BYTE " ",0
00000028 20 00
00000000
                             .code
00000000
                            main proc
00000000 B8 0000000A
                                    mov eax,10
00000005
                            start:
00000005 60
                                    pushad
00000006 BB 00000000
                                           mov ebx,0
0000000B BA 0000001
                                           mov edx,1
00000010 BE 00000000 R
                                    mov esi,OFFSET fib
00000015 89 16
                                    mov [esi],edx
00000017 83 C6 04
                                    add esi,4
0000001A B9 00000006
                                           mov ecx,6
000001F
                                    L1:
0000001F 8B C3
                                           mov eax,ebx
00000021 03 C2
                                           add eax,edx
00000023 89 06
                                           mov [esi],eax
00000025 83 C6 04
                                           add esi,4
00000028 8B DA
                                           mov ebx,edx
0000002A 8B D0
                                           mov edx,eax
0000002C E2 F1
                                    loop L1
0000002E 61
                                    popad
0000002F 60
                                    pushad
00000030 BB 00000000
                                           mov ebx,0
00000035 BE 00000000 R
                                    mov esi, OFFSET fib
0000003A BA 00000020 R
                                    mov edx, OFFSET fibStart
0000003F E8 00000000 E
                                    call writeString
00000044 8B C3
                                    mov eax,ebx
00000046 83 C0 01
                                    add eax,1
00000049 E8 00000000 E
                                    call writeInt
0000004E BA 00000025 R
                                    mov edx, OFFSET fibEnd
00000053 E8 00000000 E
                                    call writeString
00000058 8B 06
                                    mov eax,[esi]
0000005A E8 00000000 E
                                    call writeInt
0000005F BA 00000028 R
                                    mov edx, OFFSET space
00000064 E8 00000000 E
                                    call writeString
00000069 43
                                    inc ebx
0000006A B9 00000006
                                           mov ecx,6
000006F
                                    L2:
0000006F BA 00000020 R
                                           mov edx,OFFSET fibStart
00000074 E8 00000000 E
                                           call writeString
00000079 8B C3
                                           mov eax,ebx
0000007B 83 C0 01
                                           add eax,1
0000007E E8 00000000 E
                                           call writeInt
00000083 BA 00000025 R
                                           mov edx, OFFSET fibEnd
00000088 E8 00000000 E
                                           call writeString
0000008D 8B 04 9E
                                           mov eax,[esi+ebx*4]
```

00000090 E8 00000000 E call writeInt

00000095 BA 00000028 R mov edx,OFFSET space

0000009A E8 00000000 E call writeString

0000009F 43 inc ebx

000000A0 E2 CD loop L2 000000A2 61 popad

 000000A3
 E8 00000000 E
 call crlf

 000000A8
 48
 dec eax

 000000A9
 83 F8 00
 cmp eax,0

000000AC OF 8F FFFFFF53 jg start

invoke ExitProcess,0

000000B9 main endp

end main

Structures and Unions:

Name Size Offset Type

CONSOLE_CURSOR_INFO				
dwSize 00000	000	DWord		
bVisible 00000				
CONSOLE_SCREEN_BUFFER_IN			16	
dwSize 00000				
dwCursorPosition	000000	004	DWord	
wAttributes 00000	800	Word		
srWindow	000000	00A	QWord	
dwMaximumWindowSize		000000	12	DWord
COORD 00000	004			
X 00000	000	Word		
Y 00000		Word		
FILETIME 00000				
loDateTime	000000	000	DWord	
hiDateTime	000000	004	DWord	
FOCUS_EVENT_RECORD	•	000000	04	
bSetFocus 00000				
FPU_ENVIRON				
controlWord			Word	
statusWord			Word	
tagWord 00000	008	Word		
instrPointerOffset			DWord	
instrPointerSelector			DWord	
operandPointerOffset			DWord	
operandPointerSelector	000000	118	Word	
INPUT_RECORD			Word	
EventType 00000		Word		
Event 00000			ord	
bKeyDown			DWord	
wRepeatCount				
wVirtualKeyCode				
wVirtualScanCode				
			vvoru	
uChar 00000		Word	Word	
UnicodeChar 00000	000000		word	
		Byte	DMand	
dwControlKeyState	000000		DWord	
dwMousePosition	000000		DWord	
dwButtonState	000000		DWord	
dwMouseControlKeyState			DWord	
dwEventFlags	000000		DWord	
dwSize 00000		DWord	DV4	
dwCommandId	000000	JUU	DWord	

bSetFocus		000000 000000	00 04 06	DWord Word Word Word
uChar	000000	0A	Word	
UnicodeChar		000000	00	Word
AsciiChar	000000	00	Byte	
dwControlKeyState		000000	0C	DWord
MENU_EVENT_RECORD			000000	04
dwCommandId			00	DWord
MOUSE_EVENT_RECORD			000000	10
dwMousePosition		000000	00	DWord
dwButtonState				DWord
dwMouseControlKeySta	ite	000000	80	DWord
dwEventFlags		000000	0C	DWord
SMALL_RECT	•	000000	80	
Left	000000	00	Word	
Top			Word	
Right	000000	04	Word	
Bottom	000000	06	Word	
SYSTEMTIME		000000	10	
wYear	000000	00	Word	
wMonth	000000	02	Word	
wDayOfWeek		000000	04	Word
wDay			Word	
wHour	000000	80	Word	
wMinute	000000	0A	Word	
wSecond	000000	OC	Word	
wMilliseconds		000000	0E	Word
WINDOW_BUFFER_SIZE_	RECORI	D	000000	04
dwSize	000000	00	DWord	

Segments and Groups:

N a m e	Size	Length	Align	Combine	e Class
FLAT	GROUP				
STACK	32 Bit	000010	00 Para	Stack	'STACK'
_DATA	32 Bit	000000	2A Para	a Public	'DATA'
_TEXT	32 Bit	000000	B9 Para	a Public	'CODE'

Name Type Value Attr

Procedures, parameters, and locals:

CloseFile P Near	00000000 FLATLength= 00000000 External STDCALL
CloseHandle	P Near 00000000 FLATLength= 00000000 External STDCALL
Clrscr P Near	00000000 FLATLength= 00000000 External STDCALL
CreateFileA P Near	00000000 FLATLength= 00000000 External STDCALL

```
CreateOutputFile . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Crlf . . . . . . . . . . . . . . . .
                     P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
DumpMem .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
DumpRegs . . . . . . . . . . . .
ExitProcess .......... P Near 00000000 FLATLength= 00000000 External STDCALL
FileTimeToDosDateTime . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
FileTimeToSystemTime . . . . . .
                            P Near
                                    00000000 FLATLength= 00000000 External STDCALL
FlushConsoleInputBuffer . . . .
                            P Near 00000000 FLATLength= 00000000 External STDCALL
FormatMessageA . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetCommandLineA ......
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetCommandTail . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleCP . . . . . . . . .
GetConsoleCursorInfo . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleMode . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . . P Near 00000000 FLATLength= 00000000 External STDCALL
GetDateTime .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetFileTime .....
GetKeyState .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetLastError . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetLocalTime . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetMaxXY . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
GetNumberOfConsoleInputEvents .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetProcessHeap . . . . . . . .
GetStdHandle . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetSystemTime .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetTextColor . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetTickCount . . . . . . . . . .
Gotoxy . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
HeapAlloc . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
HeapCreate . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
HeapDestroy .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
HeapFree . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
                             00000000 FLATLength= 00000000 External STDCALL
HeapSize . . . . . . . . P Near
IsDigit . . . . . P Near
                             00000000 FLATLength= 00000000 External STDCALL
                             00000000 FLATLength= 00000000 External STDCALL
LocalFree . . . . . . P Near
MessageBoxA .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
MsgBoxAsk .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
MsgBox . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
OpenInputFile .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ParseDecimal32 . . . . . . . . .
ParseInteger32 . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
PeekConsoleInputA .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Random32 . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
RandomRange .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Randomize .....
ReadChar . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
ReadConsoleInputA . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ReadDec . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
ReadFile . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
ReadFloat ...... P Near 00000000 FLATLength= 00000000 External STDCALL
```

	P Near 00000000 FLATLength= 00000000 External STDCALL
ReadHex P Near	00000000 FLATLength= 00000000 External STDCALL
ReadInt P Near	00000000 FLATLength= 00000000 External STDCALL
ReadKeyFlush	P Near 00000000 FLATLength= 00000000 External STDCALL
•	<u> </u>
-	
_	P Near 00000000 FLATLength= 00000000 External STDCALL
	P Near 00000000 FLATLength= 00000000 External STDCALL
	P Near 00000000 FLATLength= 00000000 External STDCALL
	•
SetConsoleScreenBufferSize	S .
SetConsoleTextAttribute	3
SetConsoleTitleA	3
SetConsoleWindowInfo	•
SetFilePointer	P Near 00000000 FLATLength= 00000000 External STDCALL
SetLocalTime	P Near 00000000 FLATLength= 00000000 External STDCALL
SetTextColor	P Near 00000000 FLATLength= 00000000 External STDCALL
ShowFPUStack	P Near 00000000 FLATLength= 00000000 External STDCALL
Sleep P Near	00000000 FLATLength= 00000000 External STDCALL
•	00000000 FLATLength= 00000000 External STDCALL
_	P Near 00000000 FLATLength= 00000000 External STDCALL
<u> </u>	00000000 FLATLength= 00000000 External STDCALL
	_
Str_trim P Near	00000000 FLATLength= 00000000 External STDCALL
Str_ucase P Near	00000000 FLATLENGTH= 00000000 External STDCALL
_	
•	P Near 00000000 FLATLength= 00000000 External STDCALL
WaitMsg P Near	00000000 FLATLength= 00000000 External STDCALL
WriteBinB P Near	00000000 FLATLength= 00000000 External STDCALL
WriteBin P Near	00000000 FLATLength= 00000000 External STDCALL
WriteCharP Near	00000000 FLATLength= 00000000 External STDCALL
	P Near 00000000 FLATLength= 00000000 External STDCALL
WriteConsoleOutputAttribute .	——————————————————————————————————————
•	P Near 00000000 FLATLength= 00000000 External STDCALL
	00000000 FLATLength= 00000000 External STDCALL
	00000000 FLATLength= 00000000 External STDCALL
	•
	P Near 00000000 FLATLength= 00000000 External STDCALL
	00000000 FLATLength= 00000000 External STDCALL
WriteInt P Near	00000000 FLATLength= 00000000 External STDCALL
WriteStackFrameName	P Near 00000000 FLATLength= 00000000 External STDCALL
WriteStackFrame	P Near 00000000 FLATLength= 00000000 External STDCALL
WriteStringP Near	00000000 FLATLength= 00000000 External STDCALL
WriteToFile P Near	00000000 FLATLength= 00000000 External STDCALL
WriteWindowsMsg	P Near 00000000 FLATLength= 00000000 External STDCALL
main P Near	00000000 TEXT Length= 000000B9 Public STDCALL
start L Near	00000005_TEXT
L1 L Near	0000001F_TEXT
L2 L Near	0000006F_TEXT
printf P Near	00000000 FLATLength= 00000000 External C
scanf P Near	00000000 FLATLength= 00000000 External C
wsprintfA P Near	00000000 FLAT Length = 00000000 External C
wspillient i Near	COCCOCCO I LITTI LETIGETT - COCCOCCO EXCETTAT C

Name Type Value Attr

@CodeSize		er	00000000h	
@DataSize		er	00000000h	
@Interface		er	00000003h	
@Model Number	er	000000	07h	
@code Text	_TEXT			
@data Text	FLAT			
@fardata? Text	FLAT			
@fardata Text	FLAT			
@stack Text	FLAT			
ALT_MASK	Numbe	er	00000003h	
CAPSLOCK_ON	Numbe	er	00000080h	
CREATE_ALWAYS	Numbe	er	00000002h	
CREATE_NEW	Numbe	er	00000001h	
CTRL_MASK			0000000Ch	
CreateFile Text				
DO_NOT_SHARE	Numbe	er	00000000h	
ENABLE_ECHO_INPUT				
ENABLE_LINE_INPUT				
ENABLE_MOUSE_INPUT				10h
ENABLE PROCESSED INPUT		Number	r 000000	
ENABLE_PROCESSED_OUTPUT		Number	r 000000	01h
ENABLE_WINDOW_INPUT				
ENABLE_WRAP_AT_EOL_OUTP				
ENHANCED_KEY				
FALSE Numbe				
FILE_APPEND_DATA				
FILE_ATTRIBUTE_ARCHIVE				20h
FILE_ATTRIBUTE_COMPRESSED				
FILE_ATTRIBUTE_DEVICE				
FILE_ATTRIBUTE_DIRECTORY				10h
FILE ATTRIBUTE ENCRYPTED.				
FILE_ATTRIBUTE_HIDDEN				
FILE_ATTRIBUTE_NORMAL				80h
FILE_ATTRIBUTE_NOT_CONTEN				00002000h
FILE ATTRIBUTE OFFLINE	_			
FILE ATTRIBUTE READONLY				01h
FILE_ATTRIBUTE_REPARSE_POI				
FILE_ATTRIBUTE_SPARSE_FILE				
FILE_ATTRIBUTE_SYSTEM			00000004h	
FILE_ATTRIBUTE_TEMPORARY				00h
FILE BEGIN				.0011
FILE_CURRENT	Numbe	r or	00000001h	
FILE_DELETE_CHILD				
FILE_END Numbe				
FILE READ DATA		er	00000001h	
FILE SHARE DELETE			00000001n	
FILE_SHARE_READ			00000004h	
FILE_SHARE_WRITE			0000000111 00000002h	
FILE_WRITE_DATA			0000000211 00000002h	
FOCUS_EVENT			0000000211 000000010h	
		- 1	OOOOOOTOII	

CODMAT MARSSACE ALLOCATE	. DITECC	,	Number	00000100h
FORMAT_MESSAGE_ALLOCATE	_			
FORMAT_MESSAGE_FROM_SYS				100011
FormatMessage	Number	Format 	iviessageA	
GENERIC_ALL				
GENERIC_EXECUTE				
GENERIC_READ	Numbe	r	-80000000n	
GENERIC_WRITE GetCommandLine	Numbe	r	40000000h	
			nmandLineA	
HANDLE Text				
HEAP_GENERATE_EXCEPTIONS				0004h
HEAP_GROWABLE HEAP_NO_SERIALIZE	Numbe	r	00000002h	
HEAP_REALLOC_IN_PLACE_ON				0010h
HEAP_ZERO_MEMORY				
IDABORTNumbe	er	000000	03h	
IDCANCEL Number	er	000000	02h	
IDCLOSE Number	er	000000	08h	
IDCONTINUE				
IDHELP Numbe				
IDIGNORE	Numbe	r	00000005h	
IDNO Numbe				
IDOK Numbe				
IDRETRY Number	er	000000	04h	
IDTIMEOUT				
IDTRYAGAIN	Numbe	r	0000000Ah	
IDYES Numbe				
INVALID_HANDLE_VALUE				00001h
KBDOWN FLAG	Numbe	r	00000001h	
KBDOWN_FLAG KEY_EVENT	Numbe	r	00000001h	
KEY_MASKS	Numbe	r	0000001Fh	
LEFT_ALT_PRESSED				
LEFT_CTRL_PRESSED	Numbe	r	00000002h	
MB_ABORTRETRYIGNORE	14dilloc	Numbei	, 0000	0002h
MB_APPLMODAL			00000000h	000211
MB_CANCELTRYCONTINUE				0006h
MB_DEFBUTTON1			00000000h	000011
MB_DEFBUTTON2			00000100h	
MB_DEFBUTTON3			00000100h	
MB_DEFBUTTON4			00000200h	
MB_BELP			00004000h	
MB ICONASTERISK			00004000h	
MB_ICONERROR			0000004011 00000010h	
				0020h
MB_ICONEXCLAMATION				0030h
MB_ICONHAND			00000010h	00401
MB_ICONINFORMATION				0040h
MB_ICONQUESTION			00000020h	
MB_ICONSTOP			00000010h	
MB_ICONWARNING			00000030h	
MB_OKCANCEL			00000001h	
MB_OK Number		000000		
MB_RETRYCANCEL			00000005h	
MB_SYSTEMMODAL			00001000h	
MB_TASKMODAL	Numbe	r	00002000h	

MB_USERICON		Number	C	00000080h
MB_YESNOCANCEL		Number	C	0000003h
MB_YESNO		Number	C	00000004h
MENU_EVENT		Number	C	0000008h
MOUSE_EVENT		Number	C	00000002h
MessageBox		Text	Message	BoxA
NULL			0000000	
NUMLOCK_ON		Number		00000020h
OPEN_ALWAYS		Number		00000004h
OPEN_EXISTING		Number		0000003h
PeekConsoleInput				soleInputA
RIGHT_ALT_PRESSED		Number		00000001h
RIGHT_CTRL_PRESSED .				0000000111 00000004h
ReadConsoleInput				soleInputA
ReadConsole			ReadCon	•
SCROLLLOCK_ON				00000040h
SHIFT_MASK		Number	(0000004011 00000010h
SHIFT_PRESSED	• •	Number		00000010h
STD_ERROR_HANDLE				0000001011
STD_INPUT_HANDLE				0000000CH
STD_OUTPUT_HANDLE				0000000An
SetConsoleTitle				
TAB				
TRUE			0000000	
TRUNCATE_EXISTING				
VK_11	Number	r	000000B	Dh
VK_12			000000B	
VK_ADD	Numbe	r	0000006	Bh
VK_ADD VK_BACK	Numbe Numbe	r r	0000006	Bh 8h
VK_ADD	Number Number	r r Number	0000006 0000000	Bh 8h 00000003h
VK_ADD	Numbe Numbe	r r Number Number	0000006 0000000 0	Bh 8h 00000003h 00000014h
VK_ADD	Number Number 	r Number Number Number Number	0000006 0000000 0	Bh 8h 00000003h 00000014h 0000000Ch
VK_ADD	Number Number	r Number Number Number Number Number	0000006 0000000 0 0	Bh 8h 00000003h 00000014h 0000000Ch 00000011h
VK_ADD	Number	r Number Number Number Number Number	0000006 0000000 0 0 0 0	Bh 8h 00000003h 00000014h 000000011h 0000006Eh
VK_ADD	Number	r Number Number Number Number Number	0000006 0000000 0 0 0 0	Bh 8h 00000003h 00000014h 0000000Ch 00000011h
VK_ADD	Number	r Number Number Number Number Number Number	0000006 0000000 C C C C	Bh 8h 00000003h 00000014h 000000011h 0000006Eh 0000006Fh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN	Number	r Number Number Number Number Number Number Number	0000006 0000000 C C C C	Bh 8h 00000003h 00000014h 000000011h 0000006Eh 0000002Eh
VK_ADD	Number Number Number Number Number	r Number Number Number Number Number Number Number	0000006 0000000 0 0 0 0 0 00000002	Bh 8h 00000003h 00000014h 00000011h 0000006Eh 0000002Eh 00000028h
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN	Number Number Number Number Number	r Number Number Number Number Number Number Number	0000006 0000000 0 0 0 0 0 00000002	Bh 8h 00000003h 00000014h 00000011h 0000006Eh 0000002Eh 00000028h
VK_ADD	Number Number	r Number Number Number Number Number Number Number r	0000006 0000000 0 0 0 0 0 0000002	Bh 8h 00000003h 00000014h 000000011h 0000006Eh 0000002Eh 0000006Fh 00000028h
VK_ADD	Number Number Number Number Number	Number Number Number Number Number Number Number Number Number	0000006 0000000 0 0 0 0 0 0000002	Bh 8h 00000003h 00000014h 00000011h 0000006Eh 0000002Eh 00000028h 3h 0000001Bh 0000002Bh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END VK_ESCAPE VK_EXECUTE	Number Number Number Number Number	Number Number Number Number Number Number Number Number Number	0000006 0000000 0 0 0 0 0 0000002	Bh 8h 00000003h 00000014h 000000011h 0000002Eh 00000028h 3h 0000001Bh 0000002Bh 9h
VK_ADD	Number Number Number Number	Number Number Number Number Number Number Number T Number	0000006 0000000 0 0 0 0 0000002 0 0000007	Bh 8h 00000003h 00000014h 000000011h 0000002Eh 00000028h 3h 0000001Bh 0000002Bh 9h Ah
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DIVIDE VK_DOWN VK_END VK_ESCAPE VK_F10 VK_F11	Number Number Number Number Number Number Number	Number Number Number Number Number Number Number Number Number	0000006 0000000 0 0 0 0 0000002 0 0000007 0000007	Bh 8h 00000003h 00000014h 000000011h 0000006Eh 0000002Eh 00000028h 3h 0000001Bh 0000002Bh 9h Ah
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DIVIDE VK_DOWN VK_END VK_EXECUTE VK_F10 VK_F11 VK_F12 VK_F14	Number Number Number Number Number Number Number	r Number Number Number Number Number Number r Number r Number	0000006 0000000 0 0 0 0 0000002 0 0000007 0000007	Bh 8h 00000003h 00000014h 000000011h 0000000Eh 0000002Eh 00000028h 3h 0000001Bh 0000001Bh 0000002Bh 9h Ah Bh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DIVIDE VK_DOWN VK_END VK_EXECUTE VK_F10 VK_F11 VK_F12 VK_F14	Number Number Number Number Number Number Number	r Number Number Number Number Number Number r Number r	0000006 0000000 0 0 0 0 0000002 0 0000007 0000007 0000007	Bh 8h 00000003h 00000014h 000000011h 0000002Eh 00000028h 3h 0000001Bh 0000002Bh 9h Ah Bh Ch
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DELETE VK_DOWN VK_END VK_ESCAPE VK_F10 VK_F11 VK_F12 VK_F14 VK_F15	Number	r Number Number Number Number Number Number r Number r Number r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 20000014h 200000011h 2000006Eh 2000002Eh 20000008H 20000002Bh 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DIVIDE VK_DOWN VK_END VK_EXECUTE VK_F10 VK_F11 VK_F12 VK_F14	Number	Number Number Number Number Number Number Number Number T	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 20000002Eh 200000028h 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END VK_ESCAPE VK_F10 VK_F11 VK_F12 VK_F14 VK_F15 VK_F16	Number	r Number Number Number Number Number Number r Number r Number r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 2000000Eh 2000002Eh 20000002Bh 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh Fh
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DELETE VK_DIVIDE VK_END VK_END VK_EXECUTE VK_F10 VK_F11 VK_F12 VK_F13 VK_F14 VK_F15 VK_F16 VK_F16	Number	r Number Number Number Number Number Number r Number r number r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 2000000Eh 2000002Eh 20000002Bh 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh Fh 00h 1h
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_END VK_END VK_ESCAPE VK_F10 VK_F11 VK_F12 VK_F13 VK_F14 VK_F15 VK_F16 VK_F17 VK_F18 VK_F19	Number	Number Number Number Number Number Number Number Number r number r r r r r r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 20000002Eh 20000002Eh 20000002Bh 3h 20000001Bh 20000002Bh 9h Ah Bh Ch Dh Eh Fh 00h 1h 22h
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DECIMAL VK_DIVIDE VK_DOWN VK_END VK_END VK_F10 VK_F11 VK_F12 VK_F13 VK_F14 VK_F15 VK_F16 VK_F17 VK_F18 VK_F19 VK_F19	Number Number	Number Number Number Number Number Number Number Number r Number r r r r r r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 2000000Eh 2000002Eh 20000002Bh 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh Fh 00h 1h 22h 00h
VK_ADD VK_BACK VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_END VK_END VK_ESCAPE VK_F10 VK_F11 VK_F12 VK_F13 VK_F14 VK_F15 VK_F16 VK_F17 VK_F18 VK_F19	Number	r Number Number Number Number Number Number r Number r r r r	0000006 0000000 0 0 0 0 0 0 0 0 0 0 0 0	Bh 8h 20000003h 200000014h 200000011h 20000002Eh 20000028h 3h 2000001Bh 2000002Bh 9h Ah Bh Ch Dh Eh Fh 00h 1h 22h 00h 33h

VK_F22 Numb		0000085h
VK_F23 Numb		0000086h
VK_F24 Numb		0000087h
VK_F2 Numb		0000071h
VK_F3 Numb		0000072h
VK_F4 Numb	er	0000073h
VK_F5 Numb	er	0000074h
VK_F6 Numb	er	0000075h
VK_F7 Numb		0000076h
VK_F8 Numb	er	0000077h
VK_F9 Numb	er	0000078h
VK_HELP Numb	er	000002Fh
VK_HOME	Number	00000024h
VK_INSERT	Number	0000002Dh
VK_LBUTTON	Number	0000001h
VK_LCONTROL		
VK LEFT Numb		00000025h
VK_LMENU		
VK_LSHIFTNumb		
VK_MENU		
VK_MULTIPLY		
VK_NEXTNumb		0000002h
VK_NUMLOCK		
VK_NUMPAD0	Number	
VK_NUMPAD1	Number	
VK_NUMPAD2	Number	
_	Number	
VK_NUMPAD3		
VK_NUMPAD4	Number	
VK_NUMPAD5	Number Number	
VK_NUMPAD6		
VK_NUMPAD7	Number	
VK_NUMPAD8	Number	
VK_NUMPAD9	Number	
VK_PAUSE	Number	
VK_PRINT Numb		0000002Ah
VK_PRIOR	Number	
VK_RBUTTON	Number	
VK_RCONTROL	Number	
VK_RETURN	Number	
VK_RIGHT	Number	
VK_RMENU	Number	
VK_RSHIFT	Number	
VK_SCROLL	Number	
VK_SEPARATER	Number	
VK_SHIFT Numb		0000010h
VK_SNAPSHOT	Number	
VK_SPACE	Number	
VK_SUBTRACT	Number	0000006Dh
VK_TAB Numb		0000009h
VK_UP Numb		0000026h
WINDOW_BUFFER_SIZE_EVEN		Number 00000004h
WriteConsoleOutputCharacter		Text WriteConsoleOutputCharacterA
WriteConsole	Text	WriteConsoleA

```
black .....
                    Number
                                   00000000h
blue . . . . . . . . . . . . . . . . .
                    Number
                                   00000001h
brown .....
                    Number
                                   00000006h
counter ......
                    DWord 0000001C DATA
Number
                                   0000003h
exit . . . . . . . . . . . . . . . . .
                    Text
                            INVOKE ExitProcess,0
fibEnd . . . . . . . . . . . . . . .
                    Byte
                            00000025 DATA
fibStart . . . . . . . . . . . .
                            00000020 DATA
                    Byte
fib . . . . . . . . . . . . . . . . .
                    DWord 00000000 DATA
gray . . . . . . . . . . . . . . . .
                    Number
                                   00000008h
                    Number
                                   00000002h
green ......
lightBlue ..... Number
                                   0000009h
lightCyan ..... Number
                                   0000000Bh
lightGray ..... Number
                                   00000007h
lightGreen . . . . . . . Number
                                   0000000Ah
                                         000000Dh
Number
lightRed . . . . . Number
                                   0000000Ch
magenta ..... Number
                                   00000005h
red .....
                    Number
                                   00000004h
space .....
                    Byte
                            00000028 DATA
white . . . . . . . . . . . . . . . .
                    Number
                                   0000000Fh
wsprintf . . . . . Text
                            wsprintfA
yellow . . . . . Number
                                   0000000Eh
```

0 Warnings 0 Errors

Source Code

```
Author:
                     Marco Martinez
;;
       Program:
                     FibonacciNumbers.asm
;;
       Date: 10/14/2018
;;
                     Write a program that uses a loop to calculate the first seven values of the
       Purpose:
Fibonacci number sequence,
                     described by the following formula: Fib(1) = 1, Fib(2) = 1, Fib(n) = Fib(n - 1) + 1
Fib(n - 2).
;;
       Software Change Record
;;
       Name
                     Date
                                   What
;;
       Marco 10/14 Baseline for FibonacciNumbers.asm
;;
       Marco 12/3
                            Adjust for reporting
;;
INCLUDE Irvine32.inc
.data
       fib DWORD 7 DUP(0)
       counter DWORD 0
       fibStart BYTE "fib(",0
       fibEnd BYTE ")=",0
       space BYTE " ",0
.code
main proc
       push eax
       mov eax, 10
       start:
              pushad
```

```
mov ebx,0
       mov edx,1
      mov esi,OFFSET fib
      mov [esi],edx
       add esi,4
       mov ecx,6
       L1:
              mov eax, ebx
              add eax,edx
              mov [esi],eax
              add esi,4
              mov ebx,edx
              mov edx,eax
       loop L1
       popad
       pushad
      mov ebx,0
      mov esi, OFFSET fib
      mov edx,OFFSET fibStart
       call writeString
      mov eax,ebx
       add eax,1
       call writeInt
       mov edx, OFFSET fibEnd
       call writeString
       mov eax,[esi]
       call writeInt
      mov edx,OFFSET space
       call writeString
       inc ebx
       mov ecx,6
       L2:
              mov edx,OFFSET fibStart
              call writeString
              mov eax,ebx
              add eax,1
              call writeInt
              mov edx, OFFSET fibEnd
              call writeString
              mov eax,[esi+ebx*4]
              call writeInt
              mov edx, OFFSET space
              call writeString
              inc ebx
       loop L2
       popad
       call crlf
       dec eax
       cmp eax,0
jg start
pop eax
invoke ExitProcess,0
```

main endp
end main

```
Microsoft Visual Studio Debug Console
                                                                        X
fib(+1)=+1 fib(+2)=+1 fib(+3)=+2 fib(+4)=+3 fib(+5)=+5 fib(+6)=+8 fib(+7)=+13
C:\Users\tehco\source\repos\AssemTemplateProject\Debug\AssemTemplateProject.exe
(process 1060) exited with code 0.
Press any key to close this window . . .
```

Pseudo Code

```
Main Module
Begin Main
       INCLUDE Irvine32.inc
       Declare array as array = \{10,20,30,40\}
       Declare counter as integer = 1
       Declare space as string = " "
       Declare msgOriginal as string = "Original array: "
       Declare msgShifted as string = "Shifted array:
       Set ecx as 0
       Do
              Save reg
              Set esi as OFFSET array
              Set ecx as LENGTHOF array
              Set edx as OFFSET msgOriginal
              Call writeString
              Set edx as OFFSET space
              Dο
                     Set eax as [esi]
                     Call writeInt
                     Call writeString
                     Set esi as esi + 4
                     Set ecx as ecx - 1
              While (ecx > 0)
              Call crlf
              Load reg
              Save reg
              Set esi as OFFSET array + 3 * 4
              Set eax as [esi]
              Set esi as OFFSET array
              Set ebx as [esi]
              Set [esi] as eax
              Set eax as ebx
              Set esi as OFFSET array+1*4
              Set ecx as LENGTHOF array-1
              Dο
```

Set ebx as [esi]

```
Set [esi] as eax
                     Set eax as ebx
                     Set edx as edx + 1
                     Set counter as edx
                     Set esi as esi + 4
                     Set ecx as ecx - 1
              While (ecx > 0)
              Load reg
              Save reg
              Set esi as OFFSET array
              Set ecx as LENGTHOF array
              Set edx as OFFSET msgShifted
              Call writeString
              Set edx as OFFSET space
              Call writeString
              Do
                     Set eax as [esi]
                     Call writeInt
                     Call writeString
                     Set esi as esi + 4
                     Set ecx as ecx + 1
             While (ecx > 0)
              Call crlf
              Load reg
              Call crlf
              Set ecx as ecx + 1
       While (ecx > 0)
End Main
```

Listing File

Microsoft (R) Macro Assembler Version 14.15.26732.1 12/06/18 12:01:52 ..\..\Documents\School Work\P310\secondProject\ASM 2b\swapElements.asm Page 1 - 1

```
Author: Marco Martinez
    ;;
                            ShiftElements.asm
            Program:
    ;;
                            10/14/2018
            Date:
    ;;
                            Using a loop and indexed addressing, write code that rotates
            Purpose:
                            the members of a 32-bit integer array forward one position. The
                            value at the end of the array must wrap around to the first
                            position.
                            For example, the array [10,20,30,40] would be transformed into
    ;;
    ;;
                            [40,10,20,30].
    ;;
            Software Change Record
    ;;
            Name
                      Date
                                    What
    ;;
                     10/14
                                    Baseline for ShiftElements.asm
    ;;
            Marco
                    12/6
                                    Implement reporting features.
    ;;
            Marco
    ;;
    INCLUDE Irvine32.inc
C; Include file for Irvine32.lib
                                    (Irvine32.inc)
C; OPTION CASEMAP: NONE
                                    ; optional: make identifiers case-sensitive
```

```
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
A0000000 0000000A
                                    array DWORD 10,20,30,40
        0000014
        000001E
        00000028
00000010 00000001
                                    counter DWORD 1
00000014 20 00
                                           space BYTE " ",0
00000016 4F 72 69 67 69
                                    msgOriginal BYTE "Original array: ",0
        6E 61 6C 20 61
        72 72 61 79 3A
        20 00
00000027 53 68 69 66 74
                                    msgShifted BYTE "Shifted array: ",0
        65 64 20 61 72
        72 61 79 3A 20
        00
00000000
                             .code
00000000
                             main proc
00000000 B9 00000000
                                           mov ecx,0
00000005
                                    program:
00000005 60
                                           pushad
00000006 BE 00000000 R
                                           mov esi,OFFSET array
0000000B B9 00000004
                                                   mov ecx,LENGTHOF array
00000010 BA 00000016 R
                                           mov edx,OFFSET msgOriginal
00000015 E8 00000000 E
                                           call writeString
0000001A BA 00000014 R
                                           mov edx,OFFSET space
000001F
                                           L1:
0000001F 8B 06
                                                   mov eax,[esi]
00000021 E8 00000000 E
                                                   call writeInt
00000026 E8 00000000 E
                                                   call writeString
0000002B 83 C6 04
                                                   add esi,4
0000002E E2 EF
                                           loop L1
00000030 E8 00000000 E
                                           call crlf
00000035 61
                                           popad
00000036 60
                                           pushad
00000037 BE 0000000C R
                                           mov esi, OFFSET array+3*4
```

C

```
0000003C 8B 06
                                           mov eax,[esi]
0000003E BE 00000000 R
                                           mov esi, OFFSET array
00000043 8B 1E
                                           mov ebx,[esi]
00000045 89 06
                                           mov [esi],eax
00000047 8B C3
                                           mov eax,ebx
00000049 BE 00000004 R
                                           mov esi, OFFSET array+1*4
0000004E B9 00000003
                                                  mov ecx, LENGTHOF array-1
00000053
                                           L2:
00000053 8B 1E
                                                  mov ebx,[esi]
00000055 89 06
                                                  mov [esi],eax
00000057 8B C3
                                                  mov eax,ebx
00000059 42
                                                  inc edx
0000005A 89 15 00000010 R
                                                  mov counter,edx
00000060 83 C6 04
                                                  add esi,4
00000063 E2 EE
                                           loop L2
00000065 61
                                           popad
00000066 60
                                           pushad
00000067 BE 00000000 R
                                           mov esi,OFFSET array
0000006C B9 00000004
                                                  mov ecx, LENGTHOF array
00000071 BA 00000027 R
                                           mov edx,OFFSET msgShifted
00000076 E8 00000000 E
                                           call writeString
0000007B BA 00000014 R
                                           mov edx,OFFSET space
00000080 E8 00000000 E
                                           call writeString
                                           L3:
00000085
00000085 8B 06
                                                  mov eax,[esi]
00000087 E8 00000000 E
                                                  call writeInt
0000008C E8 00000000 E
                                                  call writeString
00000091 83 C6 04
                                                  add esi,4
00000094 E2 EF
                                           loop L3
00000096 E8 00000000 E
                                           call crlf
0000009B 61
                                           popad
0000009C E8 00000000 E
                                           call crlf
000000A1 41
                                           inc ecx
000000A2 83 F9 04
                                           cmp ecx,4
000000A5 OF 8C FFFFF5A
                                    jl program
                                    invoke ExitProcess,0
000000AB 6A 00
                               push +000000000h
000000AD E8 00000000 E *
                              call ExitProcess
                            main endp
000000B2
                            end main
```

Structures and Unions:

n a m e	Size	
	Offset	Type
ONSOLE_CURSOR_	INFO	

dwControlKeyState

dwMousePosition

dwButtonState

dwEventFlags

dwCommandId

dwSize 00000000

dwMouseControlKeyState 00000008

Onset	Type			
CONSOLE_CURSOR_INFO)00)04	DWord DWord		
CONSOLE_SCREEN_BUFFER_INF	0	000000)16	
dwSize 000000	000	DWord		
dwCursorPosition	000000	004	DWord	
wAttributes 000000				
srWindow	000000	00A	QWord	
dwMaximumWindowSize		000000)12	DWord
COORD 000000	004			
X 000000	000	Word		
Y 000000				
FILETIME 000000				
loDateTime	000000	000	DWord	
hiDateTime				
FOCUS_EVENT_RECORD				
bSetFocus 000000				
FPU_ENVIRON				
controlWord	000000	000	Word	
statusWord				
tagWord 000000	008	Word		
instrPointerOffset			DWord	
instrPointerSelector	000000)10	DWord	
operandPointerOffset	000000		DWord	
operandPointerSelector	000000)18	Word	
INPUT_RECORD	000000)14		
EventType 000000		Word		
Event 000000		XmmW	ord '	
bKeyDown	000000	000	DWord	
wRepeatCount	000000		Word	
wVirtualKeyCode			Word	
wVirtualScanCode		800		
uChar 000000		Word		
UnicodeChar	000000	000	Word	
AsciiChar 000000	000	Byte		

000000C

00000000

00000004

000000C

00000000

DWord

DWord

DWord

DWord

DWord

DWord

DWord

bSetFocus		000000 000000	00 04 06	DWord Word Word Word
uChar	000000	0A	Word	
UnicodeChar		000000	00	Word
AsciiChar	000000	00	Byte	
dwControlKeyState		000000	0C	DWord
MENU_EVENT_RECORD			000000	04
dwCommandId		000000	00	DWord
MOUSE_EVENT_RECORD		•	000000	10
dwMousePosition		000000	00	DWord
dwButtonState				DWord
dwMouseControlKeySta	ite	000000	80	DWord
dwEventFlags		000000	0C	DWord
SMALL_RECT	•	000000	80	
Left	000000	00	Word	
Top			Word	
Right	000000	04	Word	
Bottom	000000	06	Word	
SYSTEMTIME		000000	10	
wYear	000000	00	Word	
wMonth	000000	02	Word	
wDayOfWeek		000000	04	Word
wDay			Word	
wHour	000000	08	Word	
wMinute	000000	0A	Word	
wSecond	000000	OC	Word	
wMilliseconds		000000	0E	Word
WINDOW_BUFFER_SIZE_	RECORI	D	000000	04
dwSize	000000	00	DWord	

Segments and Groups:

N a m e	Size	Length	Align	Combine	e Class
FLAT	GROUP				
STACK	32 Bit	000010	00 Para	Stack	'STACK'
_DATA	32 Bit	000000	37 Para	a Public	'DATA'
_TEXT	32 Bit	000000	B2 Para	a Public	'CODE'

Name Type Value Attr

Procedures, parameters, and locals:

CloseFile P Near	00000000 FLATLength= 00000000 External STDCALL
CloseHandle	P Near 00000000 FLATLength= 00000000 External STDCALL
Clrscr P Near	00000000 FLATLength= 00000000 External STDCALL
CreateFileA P Near	00000000 FLATLength= 00000000 External STDCALL

```
CreateOutputFile . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Crlf . . . . . . . . . . . . . . . .
                     P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
DumpMem .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
DumpRegs . . . . . . . . . . . .
ExitProcess .......... P Near 00000000 FLATLength= 00000000 External STDCALL
FileTimeToDosDateTime . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
FileTimeToSystemTime . . . . . .
                            P Near
                                    00000000 FLATLength= 00000000 External STDCALL
FlushConsoleInputBuffer . . . .
                            P Near 00000000 FLATLength= 00000000 External STDCALL
FormatMessageA . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetCommandLineA ......
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetCommandTail . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleCP . . . . . . . . .
GetConsoleCursorInfo . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleMode . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . . P Near 00000000 FLATLength= 00000000 External STDCALL
GetDateTime .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetFileTime ......
GetKeyState .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetLastError . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetLocalTime . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetMaxXY . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
GetNumberOfConsoleInputEvents .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetProcessHeap . . . . . . . .
GetStdHandle . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetSystemTime .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetTextColor . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
GetTickCount . . . . . . . . . .
Gotoxy . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
HeapAlloc . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
HeapCreate . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
HeapDestroy .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
HeapFree . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
                             00000000 FLATLength= 00000000 External STDCALL
HeapSize . . . . . . . . P Near
IsDigit . . . . . P Near
                             00000000 FLATLength= 00000000 External STDCALL
                             00000000 FLATLength= 00000000 External STDCALL
LocalFree . . . . . . P Near
MessageBoxA .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
MsgBoxAsk .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
MsgBox . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
OpenInputFile .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ParseDecimal32 . . . . . . . . .
ParseInteger32 . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
PeekConsoleInputA .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Random32 . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
RandomRange .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Randomize .....
ReadChar . . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
P Near 00000000 FLATLength= 00000000 External STDCALL
ReadConsoleInputA . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ReadDec . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
ReadFile . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
ReadFloat ...... P Near 00000000 FLATLength= 00000000 External STDCALL
```

```
ReadFromFile . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ReadHex ...... P Near 00000000 FLATLength= 00000000 External STDCALL
ReadInt ...... P Near 00000000 FLATLength= 00000000 External STDCALL
ReadKeyFlush . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ReadKey ...... P Near 00000000 FLATLength= 00000000 External STDCALL
ReadString . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleCursorInfo . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleCursorPosition . . . .
                            P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleMode . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleScreenBufferSize . . . P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleTextAttribute . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleTitleA . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
SetConsoleWindowInfo . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
SetFilePointer . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
SetTextColor . . . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
ShowFPUStack . . . . . . . . . .
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Sleep ...... P Near 00000000 FLATLength= 00000000 External STDCALL
StrLength . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
Str compare .....
Str copy . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
Str_length . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
Str trim . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
Str_ucase ...... P Near 00000000 FLATLength= 00000000 External STDCALL
SystemTimeToFileTime . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WaitMsg ...... P Near 00000000 FLATLength= 00000000 External STDCALL
WriteBinB . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteBin . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteChar ...... P Near 00000000 FLATLength= 00000000 External STDCALL
WriteConsoleA .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
                                    P Near 00000000 FLATLength= 00000000 External STDCALL
WriteConsoleOutputAttribute ...
WriteConsoleOutputCharacterA . .
                                    P Near 00000000 FLATLength= 00000000 External STDCALL
WriteDec . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteFile . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteFloat . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
WriteHexB .....
WriteHex . . . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteInt . . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteStackFrameName . . . . . P Near 00000000 FLATLength= 00000000 External STDCALL
WriteStackFrame .....
                             P Near 00000000 FLATLength= 00000000 External STDCALL
WriteString ..........P Near 00000000 FLATLength= 00000000 External STDCALL
WriteToFile ..........P Near 00000000 FLATLength= 00000000 External STDCALL
                             P Near 00000000 FLATLength= 00000000 External STDCALL
WriteWindowsMsg . . . . . . . .
main . . . . . P Near 00000000 TEXT
                                                  Length= 000000B2 Public STDCALL
 program ..... L Near 00000005 TEXT
 L1 . . . . . . . . . . . . . . .
                     L Near 0000001F TEXT
 L2 . . . . . . . . . . . . . .
                     L Near 00000053 TEXT
                     L Near 00000085 TEXT
 L3 . . . . . . . . . . . . . .
                     P Near 00000000 FLATLength= 00000000 External C
printf . . . . . . . . . . . . . . . .
scanf ...... P Near 00000000 FLATLength= 00000000 External C
wsprintfA . . . . . . P Near 00000000 FLATLength= 00000000 External C
```

Name Type Value Attr

@CodeSize		er	00000000h	
@DataSize	Numbe		00000000h	
@Interface		er	00000003h	
@Model Numbe	er	000000	07h	
@code Text	_TEXT			
@data Text	FLAT			
@fardata? Text	FLAT			
@fardata Text	FLAT			
•	FLAT			
ALT_MASK	Numbe	er	00000003h	
CAPSLOCK_ON		er	00000080h	
CREATE_ALWAYS	Numbe	er	00000002h	
CREATE_NEW	Numbe	er	00000001h	
CTRL_MASK	Numbe	er	0000000Ch	
CreateFile Text	Create	FileA		
DO_NOT_SHARE			00000000h	
ENABLE_ECHO_INPUT			00000004h	
ENABLE_LINE_INPUT				
ENABLE_MOUSE_INPUT		Number	r 00000	010h
ENABLE_PROCESSED_INPUT		Number	r 00000	001h
ENABLE_PROCESSED_OUTPUT				001h
ENABLE_WINDOW_INPUT				008h
ENABLE_WRAP_AT_EOL_OUTP				002h
ENHANCED_KEY	Numbe	er	00000100h	
FALSE Number				
FILE_APPEND_DATA				
FILE_ATTRIBUTE_ARCHIVE	•	Numbe	r 00000	020h
FILE_ATTRIBUTE_COMPRESSED)	Numbe	r 00000	800h
FILE_ATTRIBUTE_DEVICE				
FILE_ATTRIBUTE_DIRECTORY				
FILE_ATTRIBUTE_ENCRYPTED .				000h
FILE_ATTRIBUTE_HIDDEN				
FILE_ATTRIBUTE_NORMAL				
FILE_ATTRIBUTE_NOT_CONTEN	_			00002000h
FILE_ATTRIBUTE_OFFLINE			00001000h	
FILE_ATTRIBUTE_READONLY .				
FILE_ATTRIBUTE_REPARSE_POI				
FILE_ATTRIBUTE_SPARSE_FILE				200h
FILE_ATTRIBUTE_SYSTEM			00000004h	_
FILE_ATTRIBUTE_TEMPORARY				100h
FILE_BEGIN			0000000h	
FILE_CURRENT				
FILE_DELETE_CHILD			00000040h	
FILE_END Number		000000		
FILE_READ_DATA			0000001h	
FILE_SHARE_DELETE			00000004h	
FILE_SHARE_READ			00000001h	
FILE_SHARE_WRITE			000000001	
FILE_WRITE_DATA			00000002h 00000002h	

FOCUS_EVENT	EOCUS EVENT	Numbo	_	0000001	Ωh	
FORMAT_MESSAGE_FROM_SYSTEM	FORMAT MESSAGE ALLOCATE	DITECTO	1	Number	000001001	
FORMATMESSAGE						•
GENERIC_ALL GENERIC_EXECUTE Number 20000000h GENERIC_READ Number 40000000h GENERIC_WRITE Number 40000000h GECCOMMANDLE Text GECCOMMANDLE HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_GENERATE_EXCEPTIONS Number 00000001h HEAP_GENERATE_EXCEPTIONS Number 0000001h HEAP_REALLOC_IN_PLACE_ONLY Number 00000003h IDABORT Number 00000003h IDCANCEL Number 00000003h IDCANCEL Number 00000008h IDCONTINUE Number 00000008h IDCONTINUE Number 00000009h IDIGNORE Number 00000007h IDIGNORE Number 00000007h IDOK Number 000000000h IDRETRY Number 00000000h IDRETRY Number 00000000h IDYES Number 00000000h Number 0000000h Number 00000000h Number 0000000h Number 00000000h Number 00000000h Number 00000000h Number 0000000h Number 00000000h Number 00000000h Number 00000000h Number 0000000h Number 0000000h Number 0000000h Number 00						
GENERIC_WRITE Number -80000000h GENERIC_WRITE Number 40000000h GECOmmandLine Text GetCommandLineA HANDLE Text DWORD HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_ACROWABLE Number 00000002h HEAP_REALLOC_IN_PLACE_ONLY Number 00000001h HEAP_ZERO_MEMORY Number 00000003h IDCANCEL Number 00000002h IDCANCEL Number 00000003h IDCANCEL Number 00000008h IDCANCEL Number 000000008h IDCANCEL Number 00000008h IDCANCEL Number 000000009h IDHELP Number 000000009h IDHELP Number 000000000h IDRETAY Number	CENERIC ALL	Numbo	romiau	1000000	1 0h	
GENERIC_WRITE Number -80000000h GENERIC_WRITE Number 40000000h GECOmmandLine Text GetCommandLineA HANDLE Text DWORD HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_ACROWABLE Number 00000002h HEAP_REALLOC_IN_PLACE_ONLY Number 00000001h HEAP_ZERO_MEMORY Number 00000003h IDCANCEL Number 00000002h IDCANCEL Number 00000003h IDCANCEL Number 00000008h IDCANCEL Number 000000008h IDCANCEL Number 00000008h IDCANCEL Number 000000009h IDHELP Number 000000009h IDHELP Number 000000000h IDRETAY Number	GENERIC_ALL	Number	_	1000000	Oh Oh	
GENERIC_WRITE Number 4000000h GetCommandLine Text GetCommandLineA HANDLE Text DWORD HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_GROWABLE Number 00000001h HEAP_NO_SERIALIZE Number 00000001h HEAP_REALLOC_IN_PLACE_ONLY Number 00000008h HEAP_ZERO_MEMORY Number 00000003h IDCANCEL Number 00000002h IDCANCEL Number 00000002h IDCANCEL Number 00000008h IDCOSTINUE Number 00000009h IDHELP Number 00000009h IDHON Number 00000007h IDOK Number 000000000h IDTINGO Number 000000000h IDTINGO Number 00000000h IDTINGOUN Number 00000000h IDTINGOUN Number 00000000h IDTINGOUN Number 00000000h IDTINGOUN Number	GENERIC_EXECUTE	Numbei	_	2000000	UN DOL	
GetCommandLine Text DWORD HANDLE Text DWORD HEAP_GENERATE_EXCEPTIONS Number 00000002h HEAP_GROWABLE Number 00000002h HEAP_RO_SERIALIZE Number 00000001h HEAP_REALLOC_IN_PLACE_ONLY Number 00000003h IDABORT Number 00000003h IDABORT Number 00000002h IDCANCEL Number 00000002h IDCLOSE Number 00000008h IDCONTINUE Number 00000009h IDIGNORE Number 00000007h IDK Number 000000000h IDTIMEOUT Number 00000000h IDTYSS Number 00000000h IDTYSS Number 00000000h KEY_EVENT Number 00000000h KEY_EVENT Number 00000000h KEY_EVENT Number 00000000h MB_ABORTRETRYIGHORE Number						
HEAP_GENERATE_EXCEPTIONS	GENERIC_WRITE	Number	ſ 	4000000	Oh	
HEAP_GENERATE_EXCEPTIONS	GetCommandLine	Text	GetCon	nmandLin	eA	
HEAP_GROWABLE						
HEAP_NO_SERIALIZE						
HEAP_REALLOC_IN_PLACE_ONLY						
HEAP_ZERO_MEMORY						
IDABORT						
IDCANCEL					8h	
IDCLOSE						
IDCONTINUE						
IDHELP	IDCLOSE Numbe	er	000000	08h		
IDIGNORE	IDCONTINUE	Number	r	0000000	Bh	
IDNO	IDHELP Numbe	er	000000	09h		
IDOK					5h	
IDOK	IDNO Numbe	er	000000	07h		
IDRETRY	IDOK Numbe	er	000000	01h		
IDTIMEOUT						
IDTRYAGAIN					00h	
IDYES	IDTRYAGAIN	Number	r	0000000	Ah	
INVALID_HANDLE_VALUE	IDYES Numbe	r	000000	06h		
KBDOWN_FLAG Number 00000001h KEY_EVENT Number 00000001h KEY_MASKS Number 00000001h LEFT_ALT_PRESSED Number 00000002h LEFT_CTRL_PRESSED Number 00000000h MB_ABORTRETRYIGNORE Number 00000000h MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 00000000h MB_DEFBUTTON1 Number 00000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_HELP Number 000004000h MB_ICONASTERISK Number 00000040h MB_ICONEXCLAMATION Number 00000010h MB_ICONHAND Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONWARNING Number 00000001h MB_ICONWARNING Number 000000001h MB_OKCANCEL Number 000000005h					00000001h	
KEY_EVENT						
LEFT_ALT_PRESSED Number 00000002h LEFT_CTRL_PRESSED Number 00000008h MB_ABORTRETRYIGNORE Number 000000000h MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 00000000h MB_DEFBUTTON1 Number 000000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_ICONASTERISK Number 00004000h MB_ICONASTERISK Number 00000010h MB_ICONEXCLAMATION Number 00000010h MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONWARNING Number 000000010h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 000000005h	KFY FVFNT	Number	r	0000000	1h	
LEFT_ALT_PRESSED Number 00000002h LEFT_CTRL_PRESSED Number 00000008h MB_ABORTRETRYIGNORE Number 000000000h MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 00000000h MB_DEFBUTTON1 Number 000000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_ICONASTERISK Number 00004000h MB_ICONASTERISK Number 00000010h MB_ICONEXCLAMATION Number 00000010h MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONWARNING Number 000000010h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 000000005h	KEY MASKS	Number	r	000000	Fh	
LEFT_CTRL_PRESSED Number 00000008h MB_ABORTRETRYIGNORE Number 00000000h MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 00000000h MB_DEFBUTTON1 Number 00000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_IEP Number 00000400h MB_ICONASTERISK Number 00000040h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000010h MB_ICONHAND Number 00000010h MB_ICONUESTION Number 00000020h MB_ICONSTOP Number 000000010h MB_ICONWARNING Number 00000001h MB_OKCANCEL Number 000000000h MB_RETRYCANCEL Number 000000005h	I FET ALT PRESSED	Number	r	0000001	2h	
MB_ABORTRETRYIGNORE Number 00000002h MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 000000006h MB_DEFBUTTON1 Number 00000100h MB_DEFBUTTON2 Number 00000200h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_HELP Number 000004000h MB_ICONASTERISK Number 00000040h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000030h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 000000040h MB_ICONQUESTION Number 000000010h MB_ICONWARNING Number 000000010h MB_ICONWARNING Number 000000010h MB_ICONWARNING Number 000000010h MB_ICONWARNING Number 000000010h MB_OKCANCEL Number 000000000h MB_OKCANCEL Number 0000000000h MB_RETRYCANCEL Number 000000005h						
MB_APPLMODAL Number 00000000h MB_CANCELTRYCONTINUE Number 00000006h MB_DEFBUTTON1 Number 00000000h MB_DEFBUTTON2 Number 00000200h MB_DEFBUTTON3 Number 00000300h MB_DEFBUTTON4 Number 00004000h MB_IEFBUTTON4 Number 000004000h MB_ICONASTERISK Number 00000010h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000030h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000020h MB_ICONWARNING Number 00000030h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 00000005h						
MB_CANCELTRYCONTINUE Number 00000006h MB_DEFBUTTON1 Number 00000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_HELP Number 000004000h MB_ICONASTERISK Number 00000010h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000030h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000020h MB_ICONWARNING Number 00000030h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 00000005h	_					
MB_DEFBUTTON1 Number 00000000h MB_DEFBUTTON2 Number 00000100h MB_DEFBUTTON3 Number 00000200h MB_DEFBUTTON4 Number 00000300h MB_HELP Number 00000400h MB_ICONASTERISK Number 00000010h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000010h MB_ICONINFORMATION Number 00000010h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000010h MB_ICONWARNING Number 00000001h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_OK Number 000000000000000000000000000000000000	_					
MB_DEFBUTTON2Number00000100hMB_DEFBUTTON3Number00000200hMB_DEFBUTTON4Number00000300hMB_HELPNumber00004000hMB_ICONASTERISKNumber00000040hMB_ICONERRORNumber00000010hMB_ICONEXCLAMATIONNumber00000030hMB_ICONHANDNumber00000010hMB_ICONINFORMATIONNumber00000020hMB_ICONQUESTIONNumber00000020hMB_ICONSTOPNumber00000030hMB_ICONWARNINGNumber00000030hMB_OKCANCELNumber000000001hMB_OKNumber000000000hMB_RETRYCANCELNumber000000005h						
MB_DEFBUTTON3Number00000200hMB_DEFBUTTON4Number00000300hMB_HELPNumber00004000hMB_ICONASTERISKNumber00000010hMB_ICONERRORNumber00000010hMB_ICONEXCLAMATIONNumber00000010hMB_ICONHANDNumber00000010hMB_ICONINFORMATIONNumber00000020hMB_ICONQUESTIONNumber00000020hMB_ICONSTOPNumber00000010hMB_ICONWARNINGNumber00000030hMB_OKCANCELNumber000000001hMB_OKNumber000000000hMB_RETRYCANCELNumber000000005h	_					
MB_DEFBUTTON4 Number 00000300h MB_HELP Number 000004000h MB_ICONASTERISK Number 00000010h MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000010h MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000020h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 00000001h MB_OK Number 000000000h MB_OK Number 000000000h MB_RETRYCANCEL Number 00000005h						
MB_HELPNumber00004000hMB_ICONASTERISKNumber00000040hMB_ICONERRORNumber00000010hMB_ICONEXCLAMATIONNumber00000030hMB_ICONHANDNumber00000010hMB_ICONINFORMATIONNumber00000020hMB_ICONQUESTIONNumber00000020hMB_ICONSTOPNumber00000010hMB_ICONWARNINGNumber00000030hMB_OKCANCELNumber000000001hMB_OKNumber000000000hMB_RETRYCANCELNumber000000005h	_					
MB_ICONASTERISKNumber00000040hMB_ICONERRORNumber00000010hMB_ICONEXCLAMATIONNumber00000030hMB_ICONHANDNumber00000010hMB_ICONINFORMATIONNumber00000020hMB_ICONQUESTIONNumber00000020hMB_ICONSTOPNumber00000010hMB_ICONWARNINGNumber00000030hMB_OKCANCELNumber000000001hMB_OKNumber000000000hMB_RETRYCANCELNumber000000005h	_					
MB_ICONERROR Number 00000010h MB_ICONEXCLAMATION Number 00000030h MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000020h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 000000005h						
MB_ICONEXCLAMATION Number 00000030h MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000020h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000030h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 000000005h						
MB_ICONHAND Number 00000010h MB_ICONINFORMATION Number 00000040h MB_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 000000000h MB_RETRYCANCEL Number 000000005h						
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_ICONQUESTION Number 00000020h MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 000000001h MB_OK Number 00000000h MB_RETRYCANCEL Number 00000005h	_					
MB_ICONSTOP Number 00000010h MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 00000001h MB_OK Number 00000000h MB_RETRYCANCEL Number 00000005h	_					
MB_ICONWARNING Number 00000030h MB_OKCANCEL Number 00000001h MB_OK Number 00000000h MB_RETRYCANCEL Number 00000005h	_					
MB_OKCANCEL Number 00000001h MB_OK Number 00000000h MB_RETRYCANCEL Number 00000005h	_					
MB_OK Number 00000000h MB_RETRYCANCEL Number 00000005h	_					
MB_RETRYCANCEL Number 00000005h	_				1h	
-	_					
MB_SYSTEMMODAL Number 00001000h	_					
	MB_SYSTEMMODAL	Numbe	r	0000100	0h	

MB_TASKMODAL	Numbei	00002000h
MB_USERICON	Number	00000080h
MB_YESNOCANCEL	. Number	0000003h
MB_YESNO		
MENU_EVENT		
_	Number	
MessageBox		MessageBoxA
NULL Num		00000000h
NUMLOCK_ON	Number	
OPEN_ALWAYS	Number	
OPEN_EXISTING		
PeekConsoleInput		PeekConsoleInputA
RIGHT_ALT_PRESSED		
RIGHT_CTRL_PRESSED	Numbei	00000004h
ReadConsoleInput	Text	ReadConsoleInputA
ReadConsole	Text	ReadConsoleA
SCROLLLOCK_ON	Numbei	00000040h
SHIFT MASK		
SHIFT PRESSED		
STD_ERROR_HANDLE		
STD_INPUT_HANDLE		
STD_OUTPUT_HANDLE		
SetConsoleTitle		
TAB Num		0000009h
TRUE Num		0000001h
TRUNCATE_EXISTING		00000005h
VK_11 Num	ber	000000BDh
_		
VK_12 Num		000000BBh
VK_ADD Num	ber	
	ber	000000BBh
VK_ADD Num VK_BACK Num VK_CANCEL	ber ber Numbei	000000BBh 0000006Bh 00000008h
VK_ADD Num VK_BACK Num VK_CANCEL	ber ber Numbei	000000BBh 0000006Bh 00000008h
VK_ADD Num VK_BACK Num VK_CANCEL	ber ber Numbei Numbei	000000BBh 0000006Bh 00000008h 00000003h 00000014h
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL	ber ber Numbei Numbei Numbei	000000BBh 0000006Bh 00000008h 00000003h 00000014h
VK_ADD	ber ber Numbei Numbei Numbei Numbei	000000BBh 0000006Bh 00000008h 00000003h 00000014h 0000000Ch
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL	ber ber Numbei Numbei Numbei Numbei	000000BBh 0000006Bh 00000008h 00000003h 00000014h 0000000Ch 00000011h
VK_ADD	ber Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 00000011h 0000006Eh 0000002Eh
VK_ADD	ber Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 000000Ch 00000011h 0000006Eh 0000002Eh 0000006Fh
VK_ADD	ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 0000001h 0000001h 000006Eh 0000002Eh 00000028h
VK_ADD	ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 00000011h 000000Eh 0000002Eh 0000006Fh 00000028h
VK_ADD	ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 00000011h 000000Eh 000000Eh 000000Eh 0000008Fh 00000028h 00000023h
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END Num VK_ESCAPE VK_EXECUTE	ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei Numbei Numbei	000000BBh 0000008h 0000003h 00000014h 0000001h 0000001h 000006Eh 0000002Eh 00000028h 00000023h 0000001Bh 0000002Bh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END Num VK_ESCAPE VK_F10 Num	ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei Numbei ber Numbei	000000BBh 0000008h 0000003h 00000014h 0000001h 0000001h 000002Eh 0000002Eh 00000028h 00000023h 0000001Bh 00000079h
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num	ber Number Number Number Number Number Number Number Number ber Number	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000000Eh 0000002Eh 00000028h 00000023h 0000001Bh 00000079h 0000007Ah
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num	ber Number Number Number Number Number Number Number Number ber Number ber ber	000000BBh 0000008h 0000003h 00000014h 0000001h 0000001h 000000Eh 0000002Eh 0000002Bh 00000023h 0000002Bh 0000002Bh 0000007Ah 0000007Bh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END Num VK_ESCAPE VK_EXECUTE VK_F10 Num VK_F11 Num VK_F13 Num	ber ber Number Number Number Number Number Number Number ber Number ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000000Eh 0000002Eh 0000002Bh 00000023h 0000002Bh 0000007Bh 0000007Ah 0000007Bh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F14 Num	ber ber Number Number Number Number Number Number ber Number ber ber ber ber ber	000000BBh 0000008h 0000003h 00000014h 0000001h 0000001h 000000Eh 0000002Eh 0000002Bh 00000023h 0000002Bh 0000002Bh 0000007Ah 0000007Bh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END Num VK_ESCAPE VK_EXECUTE VK_F10 Num VK_F11 Num VK_F13 Num	ber ber Number Number Number Number Number Number ber Number ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000000Eh 0000002Eh 0000002Bh 00000023h 0000002Bh 0000007Bh 0000007Ah 0000007Bh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END Num VK_ESCAPE VK_EXECUTE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F15 Num VK_F16 Num	ber ber Number Number Number Number Number Number Number ber Number ber ber ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 0000001h 0000001h 000002Eh 0000002Eh 00000028h 00000023h 0000002Bh 0000007Bh 0000007Ph 0000007Dh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_DECIMAL VK_DELETE VK_DIVIDE VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F14 Num VK_F15 Num	ber ber Number Number Number Number Number Number Number ber Number ber ber ber ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000002Eh 0000002Eh 0000002Bh 0000002Bh 0000002Bh 0000007Bh 0000007Bh 0000007Dh 0000007Dh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CAPITAL VK_CLEAR VK_CONTROL VK_DECIMAL VK_DELETE VK_DIVIDE VK_DOWN VK_END Num VK_ESCAPE VK_EXECUTE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F15 Num VK_F16 Num	ber ber Number Number Number Number Number Number ber Number ber ber ber ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000000Eh 0000002Eh 0000002Bh 0000002Bh 0000002Bh 0000007Bh 0000007Bh 0000007Ch 0000007Ch 0000007Ch 0000007Ch
VK_ADD Num VK_BACK Num VK_CANCEL VK_CLEAR VK_DECIMAL VK_DELETE VK_DOWN VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F14 Num VK_F15 Num VK_F17 Num VK_F18 Num	ber ber Number Number Number Number Number Number Number ber ber ber ber ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 0000001h 0000011h 000002Eh 0000002Eh 0000002Bh 0000002Bh 0000002Bh 0000007Bh 0000007Ph 0000007Ch 0000007Ch 0000007Ch 0000007Ch 0000007Fh 0000007Fh 0000007Fh
VK_ADD Num VK_BACK Num VK_CANCEL VK_CLEAR VK_DECIMAL VK_DELETE VK_DOWN VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F14 Num VK_F15 Num VK_F16 Num VK_F18 Num VK_F19 Num	ber ber Numbei Numbei Numbei Numbei Numbei Numbei Numbei ber	000000BBh 0000008h 0000008h 00000014h 00000011h 000000Eh 000000Eh 0000002Eh 0000002Bh 0000002Bh 0000002Bh 0000007Bh 0000007h 0000007h 000007Ch 000007Ch 000007Ch 000007Ch 0000007Ch
VK_ADD Num VK_BACK Num VK_CANCEL VK_CLEAR VK_DECIMAL VK_DELETE VK_DOWN VK_END Num VK_ESCAPE VK_F10 Num VK_F11 Num VK_F12 Num VK_F13 Num VK_F14 Num VK_F15 Num VK_F17 Num VK_F18 Num	ber ber Number Number Number Number Number Number Number ber ber ber ber ber ber ber ber ber	000000BBh 0000008h 0000008h 00000014h 0000001h 0000001h 000002Eh 0000002Eh 00000028h 00000028h 00000028h 00000079h 0000007h 000007Dh 0000007Dh 0000007Fh 0000007Fh 00000081h

VK_F21 Numbe		0084h
VK_F22 Numbe	er 00000	0085h
VK_F23 Number	er 00000	0086h
VK_F24 Numbe	er 00000	0087h
VK_F2 Number	er 00000	0071h
VK_F3 Number	er 00000	0072h
VK_F4 Numbe	er 00000	0073h
VK_F5 Numbe	er 00000	0074h
VK F6 Numbe	er 00000	0075h
VK_F7 Numbe	er 00000	0076h
VK_F8 Numbe		0077h
		0078h
VK HELP Numbe		002Fh
VK_HOME		00000024h
VK_INSERT		0000002Dh
VK_LBUTTON		
VK LCONTROL		
VK_LEFT Number		0025h
VK_LMENU		000000A4h
VK_LSHIFTNumbe		00A0h
VK_MENU		0000012h
VK_MULTIPLY		
VK_NEXTNumbe		
_		0022h
VK_NUMLOCK	Number	0000090h
VK_NUMPAD0	Number	0000060h
VK_NUMPAD1	Number	00000061h
VK_NUMPAD2	Number	00000062h
VK_NUMPAD3	Number	00000063h
VK_NUMPAD4	Number	0000064h
VK_NUMPAD5	Number	00000065h
VK_NUMPAD6	Number	00000066h
VK_NUMPAD7	Number	0000067h
VK_NUMPAD8	Number	00000068h
VK_NUMPAD9	Number	00000069h
VK_PAUSE	Number	00000013h
VK_PRINT Numbe	er 00000	002Ah
VK_PRIOR	Number	00000021h
VK_RBUTTON	Number	00000002h
VK_RCONTROL	Number	000000A3h
VK_RETURN	Number	000000Dh
VK_RIGHT	Number	00000027h
VK_RMENU	Number	000000A5h
VK_RSHIFT	Number	000000A1h
VK_SCROLL	Number	00000091h
VK_SEPARATER	Number	000006Ch
VK_SHIFT Numbe	er 00000	0010h
VK_SNAPSHOT	Number	0000002Ch
VK_SPACE	Number	00000020h
VK_SUBTRACT	Number	000006Dh
VK_TAB Numbe		0009h
VK_UP Number		0026h
WINDOW_BUFFER_SIZE_EVEN		
WriteConsoleOutputCharacter		WriteConsoleOutputCharacterA
White console output character	ΙΕΛί	WhiteconsolcoutputcharacterA

```
WriteConsole . . . . . . . . .
                          Text
                                  WriteConsoleA
array ......
                   DWord 00000000 DATA
                   Number
black .....
                                  00000000h
blue . . . . . . . . . . . . . . . .
                   Number
                                  00000001h
brown ..... Number
                                  00000006h
counter ..... DWord 00000010 DATA
cvan . . . . . . . . . . . . . . . .
                   Number
                                  0000003h
exit . . . . . . . . . . . . . . . .
                   Text
                           INVOKE ExitProcess,0
                   Number
gray . . . . . . . . . . . . . . . .
                                  00000008h
green ..... Number
                                  00000002h
lightBlue ..... Number
                                  00000009h
lightCyan ..... Number
                                  0000000Bh
lightGray ..... Number
                                  00000007h
lightGreen . . . . . . . . Number
                                  0000000Ah
000000Dh
                          Number
lightRed . . . . . Number
                                  0000000Ch
magenta ..... Number
                                  00000005h
msgOriginal .....
                                  00000016 DATA
                          Byte
msgShifted . . . . . . . . . . . . Byte
                           00000027 DATA
red .....
                   Number
                                  00000004h
                           00000014 DATA
space ......
                   Bvte
white ..... Number
                                  0000000Fh
wsprintf . . . . . Text
                           wsprintfA
                                 0000000Eh
yellow . . . . . Number
```

0 Warnings 0 Errors

Assembly Source Code

```
Author:
                     Marco Martinez
;;
                     ShiftElements.asm
       Program:
;;
       Date:
                     10/14/2018
;;
                     Using a loop and indexed addressing, write code that rotates the members
       Purpose:
;;
                     of a 32-bit integer array forward one position. The value at the end of
;;
                     the array must wrap around to the first position.
;;
                     For example, the array [10,20,30,40] would be transformed into
;;
                     [40,10,20,30].
;;
;;
       Software Change Record
;;
       Name
                   Date
                                    What
;;
       Marco
                10/14
                                    Baseline for ShiftElements.asm
;;
                            Implement reporting features.
       Marco
                12/6
;;
;;
INCLUDE Irvine32.inc
.data
       array DWORD 10,20,30,40
       counter DWORD 1
       space BYTE " ",0
       msgOriginal BYTE "Original array: ",0
       msgShifted BYTE "Shifted array: ",0
.code
main proc
       mov ecx,0
```

```
program:
              pushad
              mov esi,OFFSET array
              mov ecx, LENGTHOF array
              mov edx, OFFSET msgOriginal
              call writeString
              mov edx, OFFSET space
              L1:
                     mov eax,[esi]
                     call writeInt
                     call writeString
                     add esi,4
              loop L1
              call crlf
              popad
              pushad
              mov esi, OFFSET array+3*4
              mov eax,[esi]
              mov esi,OFFSET array
              mov ebx,[esi]
              mov [esi],eax
              mov eax,ebx
              mov esi, OFFSET array+1*4
              mov ecx, LENGTHOF array-1
              L2:
                     mov ebx,[esi]
                     mov [esi],eax
                     mov eax, ebx
                     inc edx
                     mov counter, edx
                     add esi,4
              loop L2
              popad
              pushad
              mov esi,OFFSET array
              mov ecx, LENGTHOF array
              mov edx, OFFSET msgShifted
              call writeString
              mov edx,OFFSET space
              call writeString
              L3:
                     mov eax,[esi]
                     call writeInt
                     call writeString
                     add esi,4
              loop L3
              call crlf
              popad
              call crlf
              inc ecx
              cmp ecx,4
       jl program
       invoke ExitProcess,0
main endp
end main
```

