

```

.386
.MODEL FLAT, STDCALL
OPTION CASEMAP: NONE
; прототипы внешних функций (процедур) описываются директивой
EXTERN,
; после знака @ указывается общая длина передаваемых параметров,
; после двоеточия указывается тип внешнего объекта Ц процедура
EXTERN GetStdHandle@4: PROC
EXTERN WriteConsoleA@20: PROC
EXTERN CharToOemA@8: PROC
EXTERN ReadConsoleA@20: PROC
EXTERN ExitProcess@4: PROC; функцией выхода из программы
EXTERN strlenA@4: PROC; функцией определения длины строки
EXTERN sprintfA: PROC; т.к. число параметров функции не
фиксировано,
; используется соглашение, согласно которому
очищает стек
; вызываемая процедура

.DATA; сегмент данных
STRN DB "Введите строку: ",13,10,0; выводимая строка, в конце
добавлены
; управляющие символы: 13 Ц возврат каретки, 10 Ц переход на новую
; строку, 0 Ц конец строки; с использованием директивы DB
; резервируется массив байтов
FMT DB "Число %d", 0; строка со списком форматов для функции
sprintfA
DIN DD ?; дескриптор ввода; директива DD резервирует память объемом
; 32 бита (4 байта), знак ? используется для неинициализированных
данных
DOUT DD ?; дескриптор вывода
BUF DB 200 dup (?); буфер для вводимых/выводимых строк длиной 200
байтов
LENS DD ?; переменная для количества выведенных символов
.CODE; сегмент кода
MAIN PROC; начало описания процедуры с именем MAIN
; перекодируем строку STRN
MOV EAX, OFFSET STRN; командой MOV значение второго операнда
; перемещается в первый, OFFSET Ц операция, возвращаемая адрес
PUSH EAX; параметры функции помещаются в стек командой PUSH
PUSH EAX
CALL CharToOemA@8; вызов функции
; перекодируем строку FMT
MOV EAX, OFFSET FMT
PUSH EAX
PUSH EAX
CALL CharToOemA@8; вызов функции
; получим дескриптор ввода
PUSH -10
CALL GetStdHandle@4
MOV DIN, EAX ; переместить результат из регистра EAX
; в ячейку памяти с именем DIN
; получим дескриптор вывода
PUSH -11
CALL GetStdHandle@4

```

```

MOV DOUT, EAX
; определим длину строки STRN
PUSH OFFSET STRN; в стек помещается адрес строки
CALL strlenA@4; длина в EAX
; вызов функции WriteConsoleA для вывода строки STRN
PUSH 0; в стек помещается 5-й параметр
PUSH OFFSET LENS; 4-й параметр
PUSH EAX; 3-й параметр
PUSH OFFSET STRN; 2-й параметр
PUSH DOUT; 1-й параметр
CALL WriteConsoleA@20
; ввод строки
PUSH 0; в стек помещается 5-й параметр
PUSH OFFSET LENS; 4-й параметр
PUSH 200; 3-й параметр
PUSH OFFSET BUF; 2-й параметр
PUSH DIN; 1-й параметр
CALL ReadConsoleA@20 ; обратите внимание: LENS больше числа
введенных
; символов на два, дополнительно введенные символы: 13 Ц возврат
каретки и
; 10 Ц переход на новую строку
; вывод полученной строки
PUSH 0
PUSH OFFSET LENS
PUSH LENS; длина вводимой строки
PUSH OFFSET BUF
PUSH DOUT
CALL WriteConsoleA@20
; вывод числа 123 в буфер BUF
PUSH 123
PUSH OFFSET FMT
PUSH OFFSET BUF
CALL wsprintfA
ADD ESP, 12; очистка стека от параметров (изменение регистра ESP
; на 3*4 = 12 байтов)
; вывод строки с числом 123
PUSH 0
PUSH OFFSET LENS
PUSH EAX
PUSH OFFSET BUF
PUSH DOUT
CALL WriteConsoleA@20
; небольшая задержка
MOV ECX, 03FFFFFFFH; помещение в регистр ECX Ц счетчик цикла Ц
; большого значения
L1: LOOP L1; цикл без тела
; выход из программы
PUSH 0; параметр: код выхода
CALL ExitProcess@4
MAIN ENDP; завершение описания процедуры с именем MAIN
END MAIN; завершение описания модуля с указанием первой выполняемой
процедуры

```