|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
|  | | |
| Практическое задание № 1 | | |
| по дисциплине «Разработка программ на Языке Ассемблер» | | |
|  | | |
| **ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА АССЕМБЛЕРА** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМ-71 |
| Бригада: | 6 |
| Студенты: | Камынин Антон |
|  |  |
|  |  |
| Преподаватели: | Еланцева Ирина Леонидовна |
|  | Петров Роман Владимирович |
|  | | |
| Новосибирск | | |
| 2019 | | |

Отчёт

1. Вычитание чисел, ввод в десятичной системе счисления (не менее 3-х знаков каждое число), вывод в шестнадцатеричной системе счисления.
2. Алгоритм:
3. ввод строк чисел и перевод их в числовой формат

ax = 0;

for (i=0; i<длина\_строки(s); i++) ax = ax\*10 + s[i] – '0';

1. вычитание чисел
2. преобразование чисел в строку и вывод на экран

.386

.MODEL FLAT, STDCALL

; прототипы внешних функций (процедур) описываются директивой EXTERN,

; после знака @ указывается общая длина передаваемых параметров,

; после двоеточия указывается тип внешнего объекта – процедура

EXTERN GetStdHandle@4: PROC

EXTERN WriteConsoleA@20: PROC

EXTERN CharToOemA@8: PROC

EXTERN ReadConsoleA@20: PROC

EXTERN ExitProcess@4: PROC; функция выхода из программы

EXTERN lstrlenA@4: PROC; функция определения длины строки

EXTERN wsprintfA: PROC; т.к. число параметров функции не фиксировано,

; используется соглашение, согласно которому очищает стек

; вызывающая процедура

.DATA; сегмент данных

STRN DB "Введите строку: ",13,10,0; выводимая строка, в конце добавлены

; управляющие символы: 13 – возврат каретки, 10 – переход на новую

; строку, 0 – конец строки; с использованием директивы DB

; резервируется массив байтов

FMT DB "Число %d", 0; строка со списком форматов для функции wsprintfA

DIN DD ?; дескриптор ввода; директива DD резервирует память объемом

; 32 бита (4 байта), знак «?» используется для неинициализированных данных

DOUT DD ?; дескриптор

X DW ?

Y dw ?

digit dw ?

sign dw 1

complite db 0

compliteAllert db "complite",13,10,0

BUF DB 200 dup (?); буфер для вводимых/выводимых строк длиной 200 байтов

LENS DD ?; переменная для количества выведенных символов

.CODE; сегмент кода

MAIN PROC; описание процедуры

MOV EAX, 10Q

; перекодируем строку STRN

MOV EAX, OFFSET STRN; командой MOV значение второго операнда

; перемещается в первый, OFFSET – операция, возвращающая 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 lstrlenA@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 первого числа

Input1: 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 – возврат каретки и

; 10 – переход на новую строку

call ToNum

cmp complite,0

je input1

mov X,AX

push offset compliteAllert

call lstrlenA@4

mov lens, eax

push 0

push offset lens

push lens

push offset compliteAllert

push dout

call writeconsoleA@20

;ввод второго числа

xor eax,eax

mov lens,eax

Input2: 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 – возврат каретки и

; 10 – переход на новую строку

call ToNum

cmp complite,0

je input2

mov Y,AX

push offset compliteAllert

call lstrlenA@4

mov lens, eax

push 0

push offset lens

push lens

push offset compliteAllert

push dout

call writeconsoleA@20

;вычитание чисел

mov ax, x

mov bx, y

sub ax,bx

;вывод чисел

call PrintResult

push 0

push offset lens

push lens

push offset BUF

push dout

call writeconsoleA@20

PUSH 0; параметр: код выхода

CALL ExitProcess@4

MAIN ENDP

ToNum PROC far

mov eax,lens

SUB eax,2

mov ecx, eax

mov di, 10

xor bx,bx

xor ax,ax

mov esi, offset buf

mov bl, [esi]

mov sign,0

mov complite,0

cmp bl, '-'

jne p

inc sign

inc esi

dec ecx

P:

Convert:

mov bl, [esi]

cmp bl, '0'

jb uncomplite

cmp bl,'9'

ja uncomplite

sub bl,'0'

mul di

add ax,bx

inc esi

LOOP Convert

inc complite

cmp sign,1

jne uncomplite

neg ax

; тело процедуры

uncomplite: RET

ToNum ENDP

PrintResult PROC far

mov sign, 0

xor edx,edx

xor ebx,ebx

xor ecx,ecx

cmp ax,0

mov esi, offset buf

JGE P

mov sign,1

neg ax

P:

mov bx, 16

div bx

cmp dx,9

jbe decc

add dx,55

decc:

jae strs

add dx,'0'

strs:

push edx

inc cx

xor edx, edx

cmp ax, 0

JNE P

cmp sign,0

je toPrint

push '-'

inc cx

mov lens, Ecx

xor EBX,EBX

toPrint:

mov lens, Ecx

Build:

pop ebx

mov [esi], bx

inc esi

LOOP Build

RET

PrintResult ENDP

END MAIN

1. Тесты

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | Результат | примечания | Номер |
| 1a23 | Возврат к вводу первого числа | Неверно введено первое число | 1 |
| 123  12s | Возврат к вводу второго числа | Неверно введено второе число | 2 |
| 531  1234 | -2BF | Вычитание чисел разной размерности | 3 |
| -123  -222 | 63 | Вычитание отрицательных чисел | 4 |