|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №6**

**«Создание макросов для ввода и вывода данных»**

**ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель:** практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение приемов разработки макроопределений.

**Задачи:**

1. Создать макросы для ввода и вывода чисел (двух, трех и четырехзначных).
2. С использованием макросов выполнить задание, соответствующее варианту.
3. Исходные данные вводятся с клавиатуры (n, c, d).
4. Результаты выводятся на экран.

**Вариант 15**

Ввести с клавиатуры последовательность из N чисел, размером в слово. Найти сумму первых K элементов введенной последовательности, удовлетворяющих условию: c <= a[i] <= d. Значение k, c, d задается с клавиатуры. Результат вывести на экран.

**Описание алгоритма макроса ввода-вывода:**

При вводе каждый разряд сначала переводится из символа в число, затем умножается на 10 в степени, соответствующей разряду, и складывается с регистром ax, где в итоге будет храниться результат.

При выводе число разбивается на разряды, которые поочередно переводятся в символ и выводятся в консоль, что сопровождается смещением курсора на одну позицию вправо на каждой итерации.

**Листинг:**

.model small

.386

.stack 100h

.data

buffer db 7 dup(0)

number dw 0

N db 0

K db 0

C dw 0

D dw 0

inputN db 'N: ', '$'

inputK db 'K: ', '$'

inputC db 'C: ', '$'

inputD db 'D: ', '$'

inputNumber db '>>> ', '$'

result db 'Result: ', '$'

sum dw 0

mPrint macro string

mov ah, 09h

mov dx, offset string

int 21h

endm

mMoveCursor macro row, column

mov ah, 2

mov dh, row

mov dl, column

mov bh, 0

int 10h

endm

mIncCursor macro

;; получение текущей позиции курсора

mov ah, 3

mov bh, 0

int 10h

;; сдвиг курсора на 1 вправо

inc dl

mov ah, 2

int 10h

endm

mNextLine macro

;; получение текущей позиции курсора

mov ah, 3

mov bh, 0

int 10h

;; сдвиг курсора на 1 вниз

inc dh

mov ah, 2

int 10h

endm

mInput macro buffer, size

LOCAL INPUT\_ONE\_DIGIT

LOCAL INPUT\_TWO\_DIGIT

LOCAL INPUT\_THREE\_DIGIT

LOCAL INPUT\_FOUR\_DIGIT

LOCAL END\_INPUT

;; ввод числа

mov buffer, size

inc buffer

mov ah, 0Ah

lea dx, buffer

int 21h

;; проверка на длину

mov ax, 0

cmp buffer+1, 1

je INPUT\_ONE\_DIGIT

cmp buffer+1, 2

je INPUT\_TWO\_DIGIT

cmp buffer+1, 3

je INPUT\_THREE\_DIGIT

cmp buffer+1, 4

je INPUT\_FOUR\_DIGIT

jmp END\_INPUT

INPUT\_ONE\_DIGIT:

mInputOneDigit buffer

jmp END\_INPUT

INPUT\_TWO\_DIGIT:

mInputTwoDigit buffer

jmp END\_INPUT

INPUT\_THREE\_DIGIT:

mInputThreeDigit buffer

jmp END\_INPUT

INPUT\_FOUR\_DIGIT:

mInputFourDigit buffer

END\_INPUT:

endm

mInputOneDigit macro buffer

;; перевод символа в число

mov al, buffer+2

sub al, 30h

endm

mInputTwoDigit macro buffer

;; получение десятка

mov al, buffer+2

sub al, 30h

mov bl, 10

mul bl

;; получениц единиц

add al, buffer+3

sub al, 30h

endm

mInputThreeDigit macro buffer

;; получение сотен

mov al, buffer+2

sub al, 30h

mov bl, 100

mul bl

;; получение десятка

mov bl, buffer+3

sub bl, 30h

xchg ax, bx

mov cl, 10

mul cl

add bx, ax

;; получениц единиц

mov ax, 0

mov al, buffer+4

sub al, 30h

add bx, ax

xchg ax, bx

endm

mInputFourDigit macro buffer

;; получение тысяч

mov al, buffer+2

sub al, 30h

mov bx, 1000

mul bx

;; получение сотен

mov bl, buffer+3

sub bl, 30h

xchg ax, bx

mov cl, 100

mul cl

add bx, ax

;; получение десятка

mov ax, 0

mov al, buffer+4

sub al, 30h

mov cl, 10

mul cl

add bx, ax

;; получениц единиц

mov al, buffer+5

sub al, 30h

add bx, ax

xchg ax, bx

endm

mOutput macro number

LOCAL OUTPUT\_ONE\_DIGIT

LOCAL OUTPUT\_TWO\_DIGIT

LOCAL OUTPUT\_THREE\_DIGIT

LOCAL OUTPUT\_FOUR\_DIGIT

LOCAL END\_OUTPUT

;; проверка числа на длину

cmp number, 10

jl OUTPUT\_ONE\_DIGIT

cmp number, 100

jl OUTPUT\_TWO\_DIGIT

cmp number, 1000

jl OUTPUT\_THREE\_DIGIT

cmp number, 10000

jl OUTPUT\_FOUR\_DIGIT

jmp END\_OUTPUT

OUTPUT\_ONE\_DIGIT:

mOutputOneDigit number

jmp END\_OUTPUT

OUTPUT\_TWO\_DIGIT:

mOutputTwoDigit number

jmp END\_OUTPUT

OUTPUT\_THREE\_DIGIT:

mOutputThreeDigit number

jmp END\_OUTPUT

OUTPUT\_FOUR\_DIGIT:

mOutputFourDigit number

END\_OUTPUT:

mIncCursor

endm

mOutputOneDigit macro number

;; перевод числа в символ и вывод

mov ax, number

mov ah, 9

add al, 30h

mov bl, 0Fh

mov cx, 1

int 10h

endm

mOutputTwoDigit macro number

;; перевод десятка в символ

mov ax, number

mov bx, 10

div bl

push ax

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод единиц в символ

pop ax

mov al, ah

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

endm

mOutputThreeDigit macro number

;; перевод сотен в симол

mov ax, number

mov bx, 10

div bl

push ax

mov ah, 0

div bl

push ax

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод десятка в символ

pop ax

mov al, ah

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод единиц в символ

pop ax

mov al, ah

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

endm

mOutputFourDigit macro number

;; получение тысяч, сотен, десятков и единиц

mov ax, number

mov bx, 10

mov dx, 0

div bx

push dx

div bl

push ax

mov ah, 0

div bl

push ax

add al, 30h ;; перевод тысяч в символ

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод сотен в симол

pop ax

mov al, ah

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод десятка в символ

pop ax

mov al, ah

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

;; смещение курсора

mIncCursor

;; перевод единиц в символ

pop ax

add al, 30h

;; вывод символа

mov ah, 9

mov bl, 0Fh

mov cx, 1

int 10h

endm

.code

Start:

mov ax, @data

mov ds, ax

sub ax, ax

; очистка экрана

mov ax, 600h

mov bh, 7

mov cx, 0

mov dx, 184Fh

int 10h

; перемещение курсора в начало экрана

mov ah, 2

mov dh, 0

mov dl, 0

mov bh, 0

int 10h

; ввод N

mPrint inputN

mInput buffer, 1

mov N, al

mNextLine

; ввод K

mPrint inputK

mInput buffer, 1

mov K, al

mNextLine

; ввод C

mPrint inputC

mInput buffer, 4

mov C, ax

mNextLine

; ввод D

mPrint inputD

mInput buffer, 4

mov D, ax

mNextLine

CYCLE:

; ввод числа

mPrint inputNumber

mInput buffer, 4

; проверка числа на условия

cmp K, 0

jg CONDITION\_1

jmp END\_CYCLE

CONDITION\_1:

cmp C, ax

jle CONDITION\_2

jmp END\_CYCLE

CONDITION\_2:

cmp ax, D

jle ADDITION

jmp END\_CYCLE

; добавление числа к сумме

ADDITION:

add sum, ax

dec K

END\_CYCLE:

dec N

mNextLine

cmp N, 0

jg CYCLE

; вывод результатов

mPrint result

mOutput sum

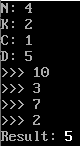
mov ax, 4C00h

int 21h

end start

END

**Результаты работы:**



**Рисунок 1.** Результат

**Вывод:** в ходе выполнения лабораторной работы были получены навыки создания макросов.