**Цель работы.**

Изучение систем счисления и средств ввода данных.

**Задание.**

1) Отладить приведённую в методических указаниях программу и закомментировать код.

2) Модифицировать программу из задания 1 для работы с трёхразрядным восьмеричным числом.

**Задание 1**

**Код программы:**

TITLE VVOD; заголовок

.Model Small; задание модели "Small"

.STACK 100h; сегмент стека в 100h байт

;Public VVOD ;закомментированная директива объявления внешней процедуры

.data; Начало сегмента данных

object dw ? ; переменная в 2 байта с неопределённым значением

message db 10,13,"Enter next bit: $" ; строка с сообщением

.code ;Начало сегмента кода

VVOD: ; метка начала

mov ax,@data

mov ds,ax; передача в DS адреса начала сегмента данных

mov object,0; инициализация object

lea di,object; передача в DI эффективного адреса object

mov cx,16; CX = 16

vv\_0\_1:

mov ah,9; AH = 9 (прерывание для вывода)

lea dx,message; передача в DX эффективного адреса message

int 21h; прерывание 21h

mov ah,1; AH = 1 (прерывание для ввода)

int 21h; прерывание 21h

cmp al,31h; сравнение кода введённого символа с кодом символа 1

jz vv1; переход на метку vv1 если условие выполнено

cmp al,30h; сравнение кода введённого символа с кодом символа 0

jz vv0; переход на метку vv0 если условие выполнено

jmp vv\_0\_1; если не введено ни 0, ни 1, начать новую итерацию

vv1:

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv0:

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

loo:

loop vv\_0\_1; цикл от метки vv\_0\_1 с счётчиком CX

mov ax,4C00h; передача в AX кода прерывания для завершения работы

int 21h; прерывание с кодом 21h

end VVOD; директива окончания программы

Новые команды, использованные в программе - clc и stc, приводящие значение CF в 0 и 1 соответственно, и rcl, выполняющая циклический сдвиг влево через перенос.

Для тестирования программы было выбрано следующее число, которое намеренно вводится с ошибками: 0110 1001 1010 0101, что в шестнадцатеричной системе счисления является числом 69 A5.

Тестирование: число 0100 0101 1101 11102= 45 DE16, намеренно допущены несколько ошибок ввода.

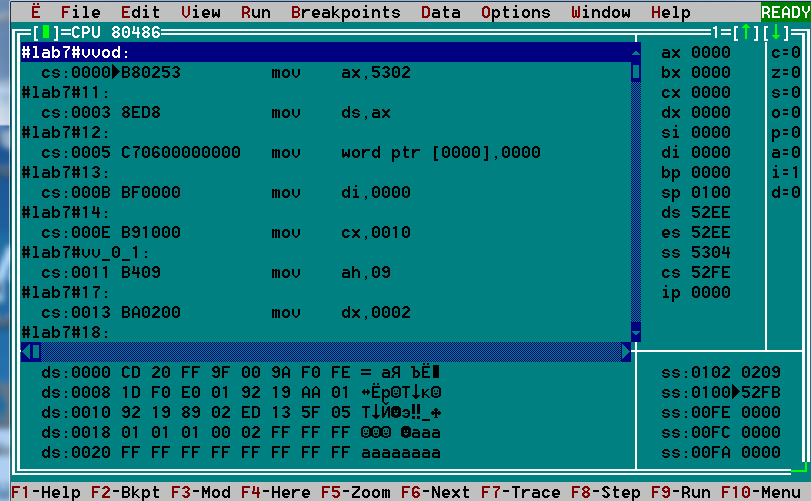


Рис. 1.1 – Программа до отладки

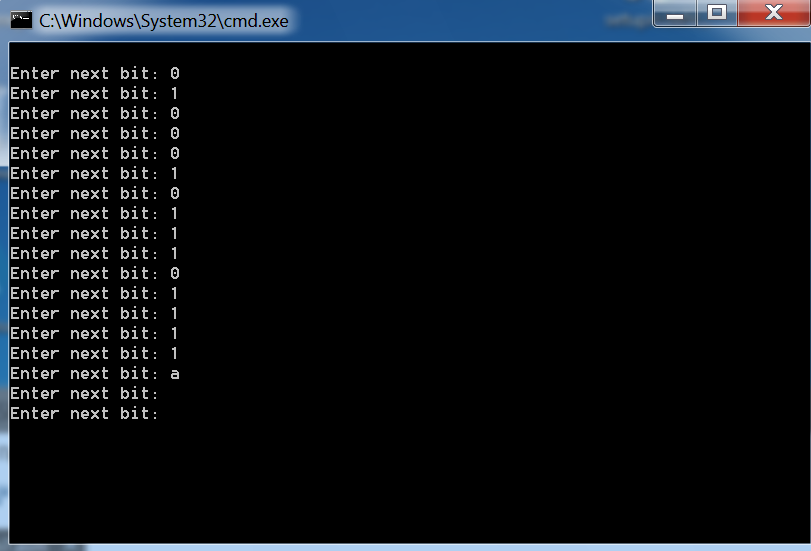


Рис. 1.2 – Побитовый ввод числа

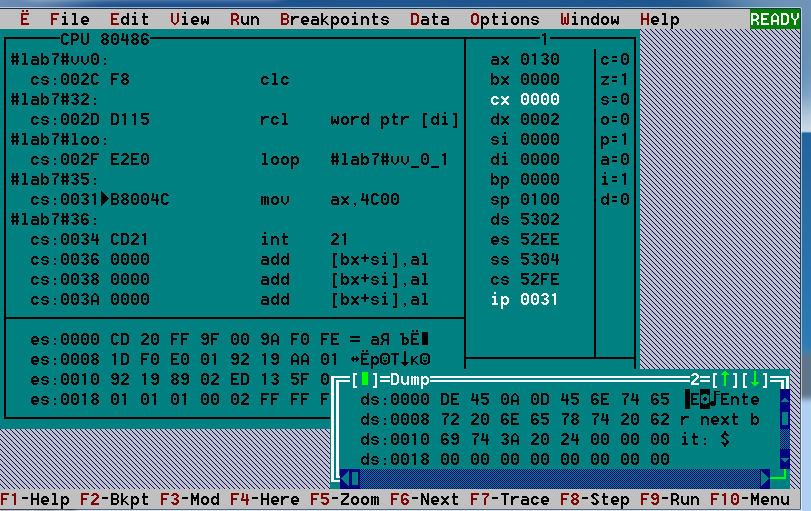


Рис. 1.3 – Программа после отладки

**Задание 2**

**Код программы:**

TITLE VVOD; заголовок

.Model Small; задание модели "Small"

.STACK 100h; сегмент стека в 100h байт

;Public VVOD ;закомментированная директива объявления внешней процедуры

.data; Начало сегмента данных

object dw ? ; переменная в 2 байта с неопределённым значением

message db 10,13,"Enter next bit: $" ; строка с сообщением

.code ;Начало сегмента кода

VVOD: ; метка начала

mov ax,@data

mov ds,ax; передача в DS адреса начала сегмента данных

mov object,0; инициализация object

lea di,object; передача в DI эффективного адреса object

mov cx,5; CX = 5 (всего 5 цифр, максимальное значение 7FFF)

; задание 16-го бита 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

vv\_0\_1\_2\_3\_4\_5\_6\_7:

mov ah,9; AH = 9 (прерывание для вывода)

lea dx,message; передача в DX эффективного адреса message

int 21h; прерывание 21h

mov ah,1; AH = 1 (прерывание для ввода)

int 21h; прерывание 21h

cmp al,31h; сравнение кода введённого символа с кодом символа 1

jz vv1; переход на метку vv1 если условие выполнено

cmp al,32h; сравнение кода введённого символа с кодом символа 2

jz vv2; переход на метку vv2 если условие выполнено

cmp al,33h; сравнение кода введённого символа с кодом символа 3

jz vv3; переход на метку vv3 если условие выполнено

cmp al,34h; сравнение кода введённого символа с кодом символа 4

jz vv4; переход на метку vv4 если условие выполнено

cmp al,35h; сравнение кода введённого символа с кодом символа 5

jz vv5; переход на метку vv5 если условие выполнено

cmp al,36h; сравнение кода введённого символа с кодом символа 6

jz vv6; переход на метку vv6 если условие выполнено

cmp al,37h; сравнение кода введённого символа с кодом символа 7

jz vv7; переход на метку vv7 если условие выполнено

cmp al,30h; сравнение кода введённого символа с кодом символа 0

jz vv0; переход на метку vv0 если условие выполнено

jmp vv\_0\_1\_2\_3\_4\_5\_6\_7; если не введено не 0-7, начать новую итерацию

loo:

loop vv\_0\_1\_2\_3\_4\_5\_6\_7; цикл от метки vv\_0\_1\_2\_3\_4\_5\_6\_7 с счётчиком CX

jmp stop

vv7:

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv6:

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv5:

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv4:

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv3:

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv2:

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv1:

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

stc; set carry flag(CF) в 1

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

vv0:

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

clc; set carry flag(CF) в 0

rcl word ptr[di],1; циклический сдвиг влево переменной object на 1 разряд и запись в младший бит значение флага CF

jmp loo; переход на метку loo

stop:

mov ax,4C00h; передача в AX кода прерывания для завершения работы

int 21h; прерывание с кодом 21h

end VVOD; директива окончания программы

Тестирование программы 1

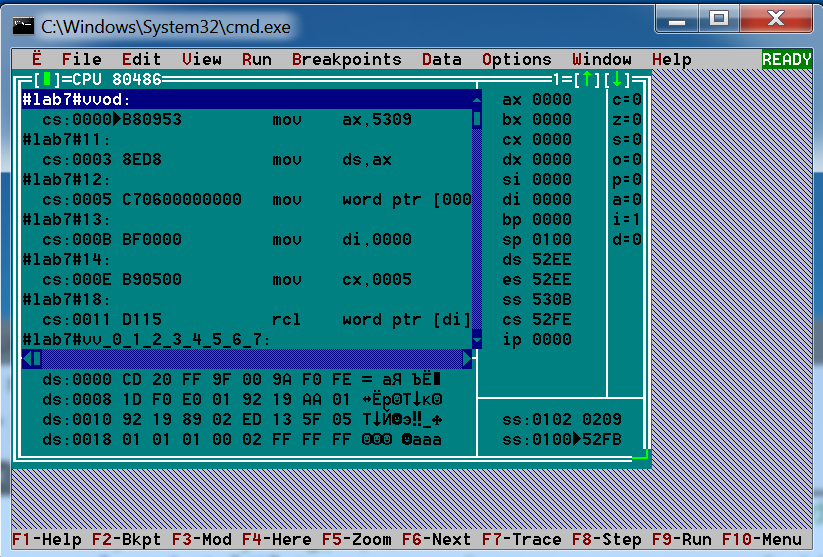


Рис. 2.1 – Программа до отладки

Для тестирования было выбрано восьмеричное число 564728 = 5D 3A16

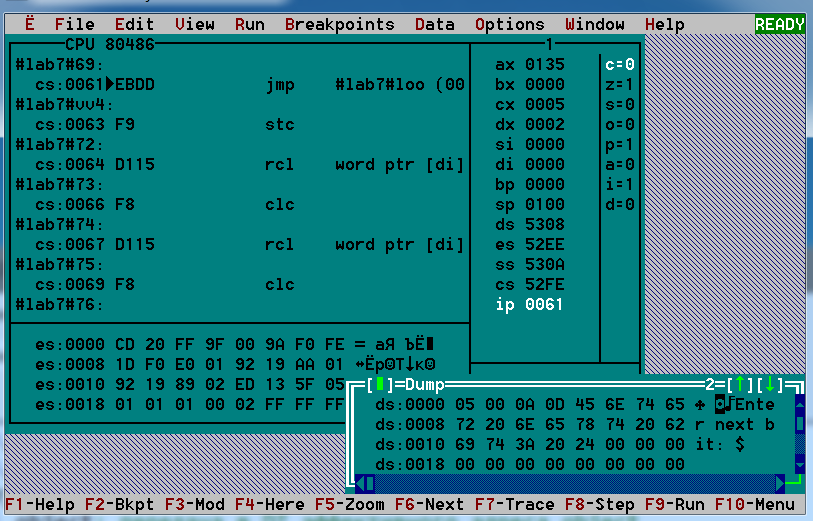


Рис. 2.2 – Обработан ввод 5

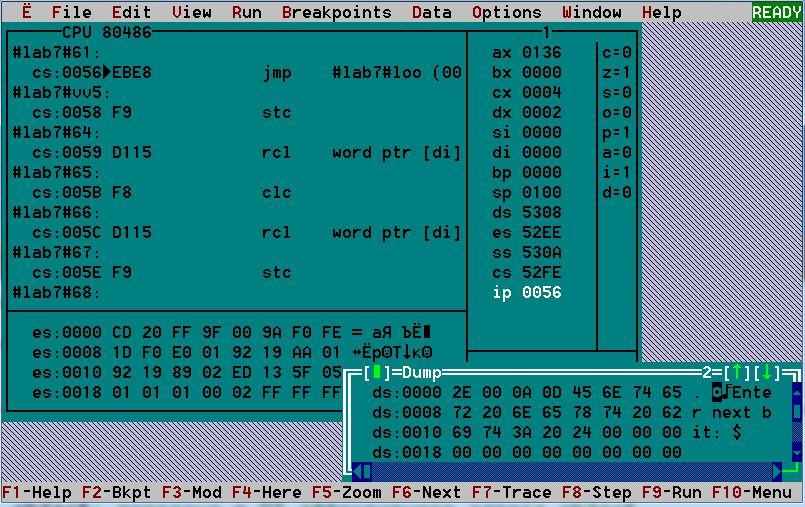


Рис. 2.3 – Обработан ввод 6 (568=2E16)

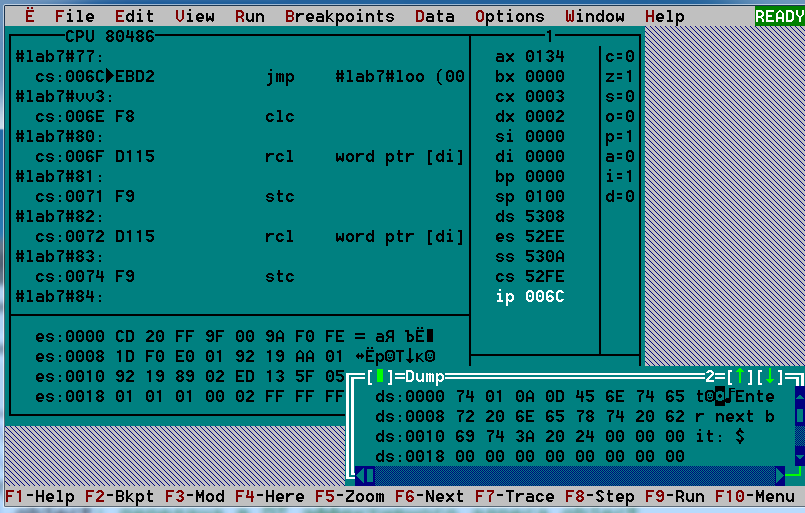


Рис. 2.4 – Обработан ввод 4 (5648=01 7416)

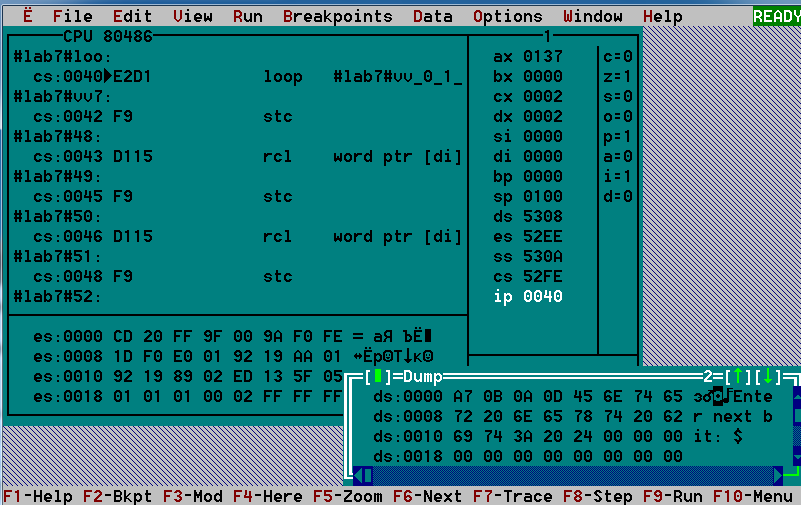


Рис. 2.5 – Обработан ввод 7 (56478=0B A716)

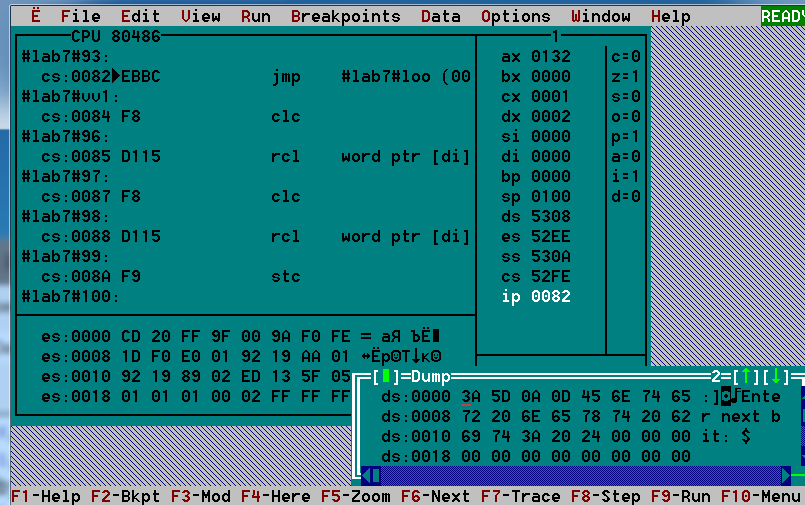


Рис. 2.6 – Обработан ввод 2 (564728=5D 3A16)

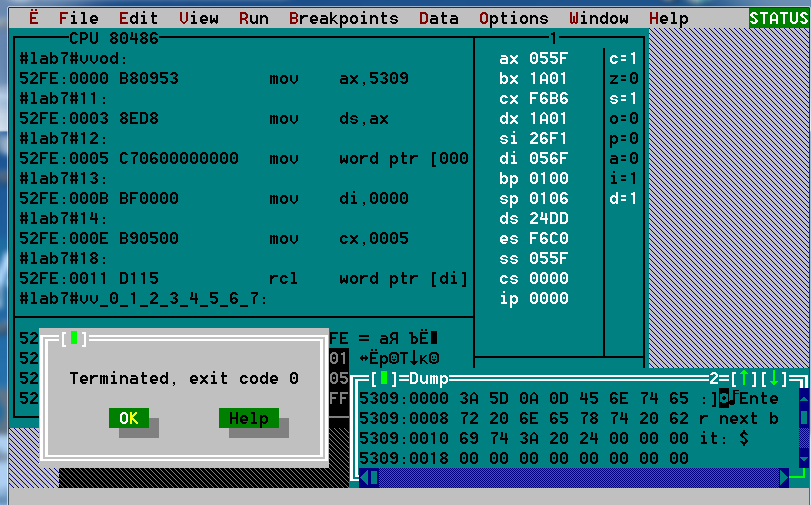


Рис. 2.7 – Программа после отладки

Тестирование программы 2

Для тестирования было выбрано восьмеричное число 777778=7F FF16, вводимое с ошибками ввода.

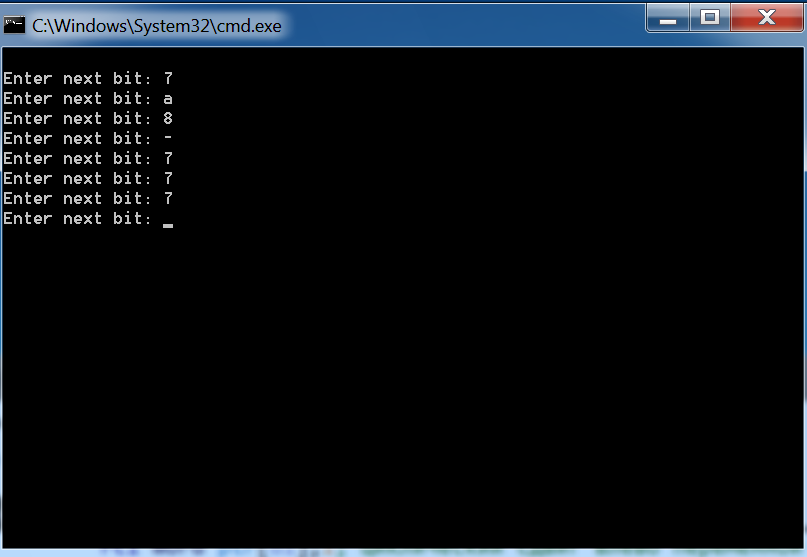
****

Рис. 2.4 – Ввод программы

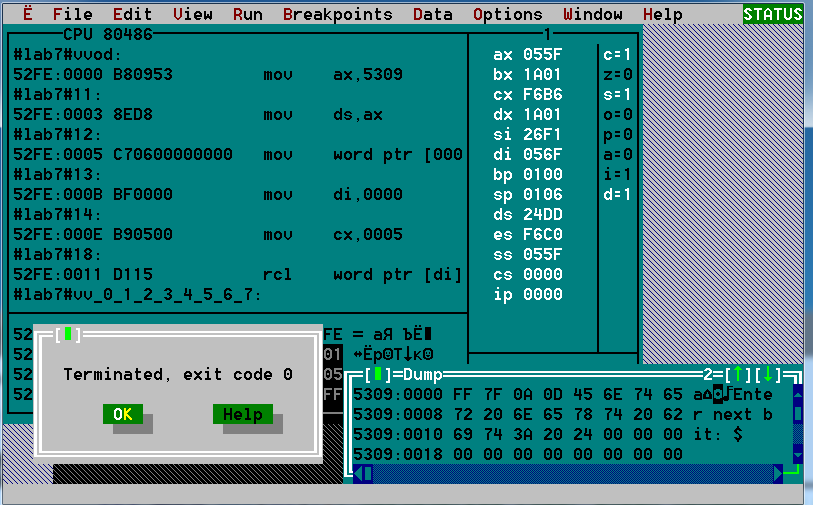
****

Рис. 2.5 – Программа после отладки

В обоих тестированиях после ввода символа производится вычитание из его кода числа 30h, так как в ASCII коды символов 0-7 начинаются от 30. В случае успешной проверки введённое число не меньше 0, но может быть больше 7, для этого производится его проверка с числом 7.

Так как любое число восьмеричной системы счисления можно представить суммой разрядов числа, умноженной на основание системы счисления в степени порядка разряда, перед добавлением каждого нового разряда предыдущие добавленные следует умножить на 8, что и осуществляется при помощи тройного логического левого сдвига. Логический сдвиг реализован через регистр CL, для чего предыдущее значение регистра CX, используемое для организации программного цикла, сохраняется в стеке и возвращается в регистр CX после выполнения сдвига.

**Выводы.**

В результате данной лабораторной работы была изучена работа с системами счисления и средствами ввода данных при помощи прерываний, освоены новые команды st[flag], cl[flag], rcl.