

ФЕДЕРАЛЬНОЕ АГЕНТСТВО  
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,  
УПРАВЛЕНИЯ И ДИЗАЙНА

Лабораторная работа № 2

по Ассемблеру на тему:

"Разработка машинно-ориентированных  
программ обработки чисел с плавающей запятой"

Вариант № 16

*Выполнил студент группы ВТ-21: Потеренко А.Г.  
Проверил преподаватель: Коноплянов А.В.*

### Порядок работы.

1. Анализ индивидуального задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке ассемблер.
3. Разработка контрольных примеров.
4. Отладка программ.
5. Составление отчета.

### Содержание отчета.

	Стр.
1. Текст постановки задачи.....	3
2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.....	3
3. Алгоритм.....	3
4 Описание и обоснование контрольных примеров.....	4

## Отчет о выполненной работе.

### 1.1. Текст постановки задачи.

Программа вводит с терминала восьмеричное число в формате:

**"ЦЕЛАЯ ЧАСТЬ.ДРОБНАЯ ЧАСТЬ"**

и выводит его в 16-ричном нормализованном виде в экспоненциальной форме. Длина мантиссы и порядка не более 5 цифр.

### 1.2. Общие требования.

- Запрещено использование MUL и DIV.
- Все ограничения по диапазону чисел касаются исходного представления данных.
- Для массивов структур обязательно представление совокупности полей структуры в одном элементе массива, то есть нельзя использовать представление массива структур через несколько массивов.
- В задачах, где от пользователя берутся численные данные, необходимо выполнить контроль допустимости этих данных.
- При вводе массивов должен быть контроль количества вводимых элементов.

### 2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.

В нашей задаче число, введенное пользователем, располагается в памяти размером в 8 байт, то есть 4 байта под целую часть и 4 байта под дробную часть. При записи вводимого значения в память мной предусмотрено – число занимает ровно столько памяти, сколько ему предназначено, например, у нас есть вводимое число "23" – оно будет занесено в память ровно в один байт, хотя мы вводим само число по цифрам. Вводя цифры, мы помещаем их в стек, затем извлекаем по цифре, записывая их в регистр, одновременно сдвигая на один бит сам регистр в сторону старшего разряда. Проводя эту операцию, мы записываем цифры, введенные пользователем в виде числа уже в память.

Так, постепенно мы записываем все число вместе с дробью в память (8 байт).

Затем, мы заносим в регистр AX четырехразрядное число, лежащее по адресу **DS:[SI]** и переводим его в шестнадцатеричное число соответствующим образом.

В конечном счете, у нас в любых двух байтах памяти будет храниться число вида: **0FFF**.

Выводим это число, соблюдая правило – байты, стоящие на четных местах выводим в виде двухразрядного числа, стоящие на нечетных – в виде одноразрядного числа.

### 3. Алгоритм поставленной задачи.

Данный алгоритм переводит число, хранящееся в памяти размером 8 байт из восьмеричной системы в шестнадцатеричную систему счисления.

#### НАЧАЛО

ЗАПИСАЛИ В DI СМЕЩЕНИЕ ПРИЕМНИКА (8 БАЙТ – S2)

ЗАПИСАЛИ В SI 0 (СМЕЩЕНИЕ ПЕРВЫХ 2-Х БАЙТ ЧИСЛА)

#### ПОВТОРИТЬ

ПОЛОЖИЛИ В РЕГИСТР AX 2 БАЙТА ИЗ ПАМЯТИ ПО АДРЕСУ [SI]

ВЫЗВАЛИ ПРОЦЕДУРУ PEREBOD (СДЕСЬ AX ПРЕОБРАЗОВЫВАЕТСЯ НУЖНЫМ ОБРАЗОМ)

ЗАПИСАЛИ 2 БАЙТА ИЗ AX ПО АДРЕСУ DS:[DI]

К SI ПРИБАВИЛИ 2 (ТЕМ САМЫМ ВЗЯЛИ СЛЕДУЮЩИЕ 2 БАЙТА ЧИСЛА)

К DI ПРИБАВИЛИ 2 (ПРОДВИГАЕМСЯ ПО S2)

ПОКА НЕ ПРОЙДЕМ ВСЕ 8 БАЙТ

#### КОНЕЦ

Алгоритм процедуры *PEREBOD*:

ТРЕБУЕМОЕ ЧИСЛО ЛЕЖИТ В AX

#### НАЧАЛО

ВЫДЕЛИЛИ ЦИФРУ ЕДИНИЦ 1-ГО БАЙТА (AL) – ПОЛОЖИЛИ В BL

ВЫДЕЛИЛИ ЦИФРУ ДЕСЯТКОВ 1-ГО БАЙТА (AL) – ЛЕЖИТ В AL

СДВИНУЛИ ВЛЕВО AL НА 1 БИТ (РАЗДЕЛИЛИ НА 2)

AL "ИЛИ" BL

ПРОДЕЛАЛИ ТУ ЖЕ ОПЕРАЦИЮ С МЛАДШИМ ПОЛУБАЙТОМ AH И СТАРШИМ ПОЛУБАЙТОМ AL

ОБЪЕДИНИЛИ БИТЫ С ПРЕДЫДУЩИМ ЗНАЧЕНИЕМ AX

ОПЯТЬ ПРОДЕЛАЛИ ТУ ЖЕ ОПЕРАЦИЮ НО УЖЕ С AH

ОПЯТЬ ОБЪЕДИНИЛИ AH С ПРЕДЫДУЩИМ ЗНАЧЕНИЕМ AX

#### КОНЕЦ

В итоге мы 3 раза сдвинули на 1 бит в сторону младших разрядов AL, старший полубайт AL и младший полубайт AH, AH.

#### 4. Тестирование программы.

№	Вводимое пользователем число	Ответ программы	+ -
1	0,00000000	0P-0	+
2	0,00000007	7,P-6	+
3	0,0000007	3,8P-5	+
4	0,000007	1,C0P-4	+
5	0,00007	E,00P-4	+
6	0,0007	7,000P-3	+
7	0,007	3,8000P-2	+
8	0,07	1,C0000P-1	+
9	0,7	E,00000P-1	+
10	77,7	3,FE00000P+1	+
11	77,07	3,F1C0000P+1	+
12	77,007	3,F038000P+1	+
13	77,0007	3,F007000P+1	+
14	77,00007	3,F000E00P+1	+
15	77,000007	3,F0001C0P+1	+
16	77,0000007	3,F000038P+1	+
17	77,00000007	3,F000007P+1	+
18	77,00000000	3,F0000000P+1	+
19	000,00000	0P-0	+
20	77,7	3,FE00000P+1	+
21	777,7	1,FFE00000P+2	+
22	7777,7	F,FFE00000P+2	+
23	77777,7	7,FFFE00000P+3	+
24	777777,7	3,FFFFE00000P+4	+
25	7777777,7	1,FFFFFF1C0000P+5	+
26	77777777,7	F,FFFFFFE00000P+5	+

```

;////////////////////////////////////
;////////////////////////////////////Лабораторная работа №2 по ассемблеру////////////////////////////////////
;////////////////////////////////////
.MODEL SMALL
.DATA
S1 DQ ? ;ПАМЯТЬ ДЛЯ ВОСЬМЕРИЧНОГО ЧИСЛА
S2 DQ ? ;ПАМЯТЬ ДЛЯ ШЕСТНАДЦАТЕРИЧНОГО ЧИСЛА
K1 DB ? ;СЧИТАЕТ СКОЛЬКО ЦИФР В ЦЕЛОЙ ЧАСТИ
K2 DB ? ;СЧИТАЕТ СКОЛЬКО ЦИФР В ДРОБНОЙ ЧАСТИ
K3 DB ? ;СЧИТАЕТ ПОРЯДОК ЭКСПОНЕНТЫ
VIDEO DB ?
ST1 DB 'BBEDITE:"+/-CHISLO,CHISLO", A UG ZATEM GMITE ENTER:',0AH,0DH,'$'
ST2 DB 'PEREBEDENNOE CHISLO:',0AH,0DH,'$'
PERENOS DB 0AH,0DH,'$'
ZNAC_CH DB ?
.STACK 100h
.CODE
MAIN:
;////////////////////////////////////
;МАКРОС, ОТВЕТСТВЕННЫЙ ЗА ВЫВОД СООБЩЕНИЙ ПОЛЬЗОВАТЕЛЮ, ЗАПОЛНЕНИЕ СЕГМЕНТОВ
;DS И ES, ПОВЕДЕНИЕ КУРСОРА И ВИДЕОРЕЖИМ (ИССЛЕДОВАНИЯ РЕЖИМА ДО НАЧАЛА ПРОГР.)
;////////////////////////////////////
MACROS_BEGIN MACRO
    MOV AX,@DATA
    MOV DS,AX
    MOV AX,0B800H
    MOV ES,AX
    ;Узнали номер текущего видеорежима
    MOV AX,4F03H
    INT 10H
    MOV VIDEO,BL ;Сохранили его номер в памяти
    ;Установили текстовый видеорежим
    MOV AH,00H
    MOV AL,2
    INT 10H
    ;ПРИГЛАШЕНИЕ КО ВВОДУ
    LEA DX,ST1
    MOV AH,9
    INT 21H
    MOV BP,162
    ;СООБЩЕНИЕ О ВЫВОДЕ ЧИСЛА
    LEA DX,PERENOS
    INT 21H
    LEA DX,ST2
    INT 21H
    ;СПРЯТАТЬ КУРСОР
    MOV AH,02H
    MOV DH,100
    MOV DL,100
    INT 10H
ENDM
;////////////////////////////////////
;////////////////////////////////////МАКРОС ЗАПОЛНЕНИЯ НУЛЯМИ ОСТАВШИХСЯ РАЗРЯДОВ В ДРОБИ////////////////////////////////////
;////////////////////////////////////
MACROS_NOL MACRO
    PUSH DX
    MOV DL,K2
    MOV DH,8
    SUB DH,DL
    MOV CL,DH ;ТЕПЕРЬ РЕГИСТР CL СОДЕРЖИТ, СКОЛЬКО 0 НУЖНО ДЛЯ ДОПОЛНЕНИЯ
    POP DX
    CMP cl,0
    JE NOU_PUSH ;ЕСЛИ ВСЕ РАЗРЯДЫ ЗАНЯТЫ, ТО НЕЗАЧЕМ ИХ ДОПОЛНЯТЬ НУЛЯМИ
GC:
    PUSH 0
    LOOP GC
ENDM
;////////////////////////////////////
;////////////////////////////////////"+" OR "-"////////////////////////////////////

```

```
;
;////////////////////////////////////
MACROS_TOCHKA MACRO
CIKL_TOCHKA:
    MOV AH,08H
    INT 21H
    CMP AL,43
    JGE OK_TOCHKA
    JMP CIKL_TOCHKA
    OK_TOCHKA:
    CMP AL,45
    JLE OK_TOCHKA@
    JMP CIKL_TOCHKA
    OK_TOCHKA@:
    CMP AL,44
    JNE OK_TOCHKA@@
    JMP CIKL_TOCHKA
JMP CIKL_TOCHKA
OK_TOCHKA@@:MOV ZNAC_CH,AL
    MOV AH,7
    MOV BP,160
    MOV ES:[BP],AX
    ADD BP,2
ENDM
;
;////////////////////////////////////
;/////////////////////////////////МАКРОС НАСТРОЙКИ////////////////////////////////////
;////////////////////////////////////
NASTROIKA MACRO P1
    LEA SI,S2
    ADD SI,P1
    MOV BL,30H
    MOV CX,4
ENDM
;
;////////////////////////////////////
;/////////////////////////////////Макрос занесения числа в память////////////////////////////////////
;////////////////////////////////////
COOL MACRO PAR1,PAR2,PAR3,PAR4,PAR5,STEK_PUST
    CMP SP,100h ;ЕСЛИ СТЕК ВООБЩЕ ПУСТОЙ
    JE STEK_PUST
    XOR CH,CH
    MOV CL,PAR1
    MOV SI,PAR3
PAR5:
    POP AX ;ИЗВЛЕКЛИ ТО, ЧТО БЫЛО В СТЕКЕ
    ;***ЕСЛИ В СТЕКЕ ОСТАЛАСЬ ОДНА ЦИФРА, ТО СРАЗУ ЗАПИСЫВАЕМ ЕЕ В ПАМЯТЬ**
    CMP CL,1
    JE PAR2
    ;*****
    MOV BL,AL
    POP AX
    SHL AL,4
    OR AL,BL
    MOV DS:[SI],AL ;В ИТОГЕ В ПАМЯТЬ ЗАПИСЫВАЕМ 2Х РАЗРЯДНОЕ ЧИСЛО
    ;*****ИЗМЕНЯЕМ НАШИ СЧЕТЧИКИ*****
    INC SI
    SUB CL,2
    ;*****ЕСЛИ СТЕК ПУСТ ВООБЩЕ, ТО ПЕРЕХОДИМ НА СЛЕДУЮЩУЮ МЕТКУ*****
    CMP CL,0
    JE PAR4
JMP PAR5
PAR2:
    MOV DS:[SI],AL
STEK_PUST:
ENDM
;
;////////////////////////////////////
;/////////////////////////////////Макрос занесения цифры в стек////////////////////////////////////
;////////////////////////////////////
COOL@ MACRO NOESC,NOPUSTOCIFR,PARTOCHKA,PARCIKL,PAROK,PAROK1,PARENTER,PAR_K?,JPP,KOS
    MOV cx,8
PARCIKL:
```

```

MOV AH,08H
INT 21H
;*****
CMP AL,27
JNE NOESC
JMP NEAR:EXIT@
NOESC:
;*****ENTER МОЖНО ЖАТЬ ТОЛЬКО ПОСЛЕ ВВОДА ДРОБИ*****
CMP AL,PARENTER
JNE JPP
JMP NEAR:PAMAT@
JPP:
;*****Ввод запятой*****
PUSH DX
MOV DL,K1 ;НАМ НУЖНО ЗНАТЬ В МАКРОСЕ ДЛЯ ЦЕЛОГО ЧИСЛА
CMP DL,0 ;БЛОК ПРОВЕРЯЕТ НЕ НАЖАТА ЛИ ПЕРВОЙ ЗАПЯТАЯ
POP DX
JE NOPUSTOCIFR
;*****
CMP AL,PARTOCHKA ;ПРИ ПОВТОРНОМ ВЫЗОВЕ ЗАПЯТАЯ УЖЕ НЕ НУЖНА, ЗАМЕНЯЕМ 1
JNE NOPUSTOCIFR
;*****
CALL ТОЧКА ;ЕСЛИ ВСЕ ЖЕ НАЖАТА ЗАПЯТАЯ И ОНА НЕ ПЕРВАЯ, ТО ВЫВОДИМ ЕЕ
JMP NEAR:PAMAT ;МЫ ЗАХОДИМ СЮДА ТОЛЬКО В ПЕРВОМ МАКРОСЕ
;*****ФИЛЬТР ЧИСЕЛ ОТ 0 ДО 7*****
NOPUSTOCIFR:
CMP AL,30H
JGE PAROK
JMP PARCIKL
PAROK:
CMP AL,37H
JLE PAROK1
JMP PARCIKL
;*****ВЫВОД НА ЭКРАН*****
PAROK1:
CALL CIFRA
;*****ЗАНОСИМ ЦИФРУ В СТЕК*****
SUB AL,30H
XOR AH,AH
PUSH AX
INC PAR_K?
LOOP PARCIKL
MOV BL,PARENTER
CMP BL,13 ;ЕСЛИ 2 МАКРОС, ТО ВВОДИТЬ ЗАПЯТУЮ НЕЗАЧЕМ
JE KOS
CALL ТОЧКА ;ЕСЛИ ПОЛЬЗОВАТЕЛЬ УМУДРИЛСЯ НЕ ВВЕСТИ ЗАПЯТУЮ
KOS:NOP
ENDM
;////////////////////////////////////
;////////////////////////////////////МАКРОС ПОИСКА НЕНУЛЕВОГО БАЙТА В ЦЕЛОЙ ИЛИ ДРОВНОЙ ЧАСТИ////////////////////////////////////
;////////////////////////////////////
POISK_NENUL_BAIT MACRO P1,P2,P3,P4
P1:
MOV AL,[SI]
CMP AL,00H
JNE P2
;ЕСЛИ МАКРОС ИСПОЛЬЗУЕТСЯ ВО 2 РАЗ, ТО ПЕЧАТАЕМ НЕДОСТАЮЩИЕ НУЛИ СЛЕВА
CMP DL,1
JNE P3
CALL СНЕТ_НЕЧЕТ ;УЗНАЕМ ПОЗИЦИЮ
JE P4
CALL ПЕЧАТ_NOL ;ЕСЛИ SI - НЕЧЕТНОЕ, ТО ПЕЧАТАЕМ ДОПОЛНИТЕЛЬНЫЙ НОЛЬ
P4: CALL ПЕЧАТ_NOL ;ПЕЧАТАЕМ НОЛЬ В НЕЗАВИСИМОСТИ ОТ SI
P3:DEC SI
LOOP P1
ENDM
;-----
;
;

```

```

;
;
;
;
;
;
;
;
;
;-----
MACROS_BEGIN ;НАСТРОЙКА ИНТЕРФЕЙСА
;-----"+" OR "-"-----
MACROS_TOCHKA
;-----РАБОТА С ЦЕЛОЙ ЧАСТЬЮ-----
COOL@ ESC1,NOPUSTOCIFR1,44,CL1,OK1,OK2,1,K1,GO_НОМ,KOS1 ;1=нельзя 0 расшир.кл.
PAMAT:COOL K1,GO_POSLEDN_CIFRA,0,GO_DROBNOE,CIKL1,STACK1
;-----РАБОТА С ДРОБНОЙ ЧАСТЬЮ-----
GO_DROBNOE:COOL@ ESC2,NOPUSTOCIFR2,1,CL2,OK3,OK4,13,K2,GO_НОМ@,KOS2
PAMAT@:MACROS_NOL
NOU_PUSH:COOL 8,GO_POSLEDN_CIFRA@,4,GO_PEREBOД,CIKL2,STACK2
;-----Перевод числа из 8-ной в 16-ю систему счисления-----
GO_PEREBOД:
    MOV CX,4
    XOR SI,SI
    XOR DI,DI
    LEA DI,S2
BEGIN:
    MOV AX,[SI]
    CALL PEREBOD
    MOV DS:[DI],AX
    ADD SI,2
    ADD DI,2
LOOP BEGIN
;-----
;-----Вывод на экран-----
;-----
MOV BP,480 ;РАБОТАЕМ С ВИДЕОРЕЖИМОМ
MOV DL,0 ;ФЛАГ, ОТВЕЧАЮЩИЙ ЗА ТО, КАКОЕ ЦЕЛОЕ-НУЛЕВОЕ ИЛИ НЕНУЛЕВОЕ
MOV AL,ZNAC_CH
MOV AH,7
MOV ES:[BP],AX
ADD BP,2
;-----
NASTROIKA 3
POISK_NENUL_BAIT CIKL_POISK,KONTAKT,HOO,NECH
;-----
MOV DL,0 ;ФЛАГ - НУЛЕВАЯ ЦЕЛАЯ ЧАСТЬ
JMP NEAR:DROBNOE ;ЕСЛИ ЦЕЛАЯ ЧАСТЬ=0, ТО ПЕРЕХОДИМ К АНАЛИЗУ ДРОБНОЙ ЧАСТИ
;-----
;-----АНАЛИЗ ЦЕЛОЙ ЧАСТИ-----
;-----
KONTAKT:
    MOV DL,1 ;УСТАНОВЛИВАЕМ ФЛАГ - ЦЕЛОЕ НЕНУЛЕВОЕ
    CMP SI,8 ;ЕСЛИ ЧИСЛО ИМЕЕТ НЕ БОЛЕЕ ДВУХ РАЗРЯДОВ- И ОНО НЕНУЛЕВОЕ
    JNE NET
    CALL SDVIG_RAZRAD_DESATOK
    JE SHEL
;-----
;ЕСЛИ РАЗРЯД 10-K<>0, ТО ЕСТЬ НЕ БУДЕТ ОТВЕТА ВИДА "0,А", А БУДЕТ "А,"
CALL PECHAT_CHEТНОЕ_DEC
CALL TOCHKA
CALL PECHAT_CHEТНОЕ_ED
INC K3 ;УВЕЛИЧИВАЕТ ПОРЯДОК ЭКСПОНЕНТЫ
JMP NEAR:DROBNOE
;-----
SHEL:;ЕСЛИ ВСЕ ЖЕ РАЗРЯД 10-K РАВЕН 0, ПЕЧАТАЕМ ТОЛЬКО НЕНУЛЕВУЮ ЧАСТЬ
CALL PECHAT_NECHEТНОЕ
CALL TOCHKA
JMP NEAR:DROBNOE

```



```
;
NET:                                     ;ВСЕ ОСТАЛЬНЫЕ СЛУЧАИ - ЕСЛИ SI<>LEA S2 И AL<>0
                                           ;ВЫВОД ПЕРВОЙ ЦИФРЫ, ЗАПЯТОЙ, ВТОРОЙ ЦИФРЫ

CALL CHET_NECHEТ
JE NECHETNOE                               ;ЕСЛИ SI ЕСТЬ НЕЧЕТНОЕ ЧИСЛО
CHETNOE:  CALL SDVIG_RAZRAD_DESATOK ;ПРОВЕРИМ РАЗРЯД 10-К - ОН ДОЛЖЕН<>0
          JE ES_DEC_RABNO_NOL
          ;-----
          CALL PECHAT_CHETNOE_DEC          ;ЕСЛИ РАЗРЯД 10-К НЕ РАВЕН НУЛЮ
          CALL ТОЧКА
          CALL PECHAT_CHETNOE_ED
          INC K3                               ;УВЕЛИЧИВАЕТ ПОРЯДОК ЭКСПОНЕНТЫ
          JMP NET_ES_DEC_RABNO_NOL
          ;-----
          ES_DEC_RABNO_NOL:CALL PECHAT_CHETNOE_ED
                           CALL ТОЧКА
          NET_ES_DEC_RABNO_NOL:NOP
          JMP NOU_NECHEТNOE
NECHETNOE:CALL PECHAT_NECHEТNOE
          CALL ТОЧКА
NOU_NECHEТNOE:                             ;ПЕЧАТЬ ОСТАЛЬНОЙ ЧАСТИ ЦЕЛОГО ЧИСЛА-БЕЗ ЗАПЯТЫХ
          DEC SI
          MOV AL,[SI]
          CALL CHET_NECHEТ
          JE NECHETNOE@                      ;ФЛАГ УСТАНОВЛИВАЕТСЯ ЕСЛИ ЧИСЛО НЕЧЕТНО
          CHETNOE@:CALL PECHAT_CHETNOE_DEC
                           INC K3
                           CALL PECHAT_CHETNOE_ED
                           INC K3
                           JMP NOU_NECHEТNOE@
          NECHETNOE@:CALL PECHAT_NECHEТNOE
                           INC K3
          NOU_NECHEТNOE@: CMP SI,8
                           JE DROBNOE

JMP NOU_NECHEТNOE
;
;-----АНАЛИЗ И ПЕЧАТЬ ДРОБИ-----
;
DROBNOE:NASTROIKA 7
POISK_NENUL_BAIT CIKL_POISK@,KONTAKT@,HO@,NECH@
CALL PECHAT_NOL                               ;ЕСЛИ ВООБЩЕ ДРОБНАЯ ЧАСТЬ СОСТОИТ ИЗ НУЛЕЙ
JMP NEAR:EXIT                               ;ПЕЧАТАЕМ НОЛЬ И ВЫХОДИМ
;
KONTAKT@:
  CMP DL,1
  JNE HOO@
  ;-----ЕСЛИ ЦЕЛОЕ - ИМЕЕТ НЕНУЛЕВОЕ ЗНАЧЕНИЕ-----
  CALL CHET_NECHEТ
  JE HOO@
  CALL SDVIG_RAZRAD_DESATOK
  JNE HOO@
  CALL PECHAT_NOL          ;ЕСЛИ ЧИСЛО, НАПРИМЕР 07, А НЕ 70-ПЕЧАЕМ НОЛЬ "0"7
  ;-----ЕСЛИ ЦЕЛОЕ ИМЕЕТ НУЛЕВОЕ ЗНАЧЕНИЕ-----
  HOO@:
  CMP SI,12
  JNE NET@                      ;ЕСЛИ ЧИСЛО ИМЕЕТ НЕ БОЛЕЕ ДВУХ РАЗРЯДОВ
  CALL SDVIG_RAZRAD_DESATOK
  JE SHEL@
  ;-----
  ;-----ЕСЛИ РАЗРЯД 10-К<>0, НЕ БУДЕТ ОТВЕТА ВИДА "0,А", А БУДЕТ "А,"-----
  ;-----
  CALL PECHAT_CHETNOE_DEC
  CMP DL,0H
  JNE SUP
  ;-----ЕСЛИ ЦЕЛОЕ ИМЕЕТ НУЛЕВОЕ ЗНАЧЕНИЕ-----
  CALL ТОЧКА
  ;-----ЕСЛИ ЦЕЛОЕ<>0, ТО ", "- НЕ НУЖНА-----
  SUP:CALL PECHAT_CHETNOE_ED
      JMP NEAR:EXIT
```

```

;-----
;---ЕСЛИ ВСЕ ЖЕ РАЗРЯД 10-К РАВЕН 0, ПЕЧАТАЕМ ТОЛЬКО НЕНУЛЕВУЮ ЧАСТЬ---
;-----
SHEL@:CALL PECHAT_NECHETNOE
CMP DL,0H
JNE SUP@
;-----ЕСЛИ ЦЕЛОЕ ИМЕЕТ НУЛЕВОЕ ЗНАЧЕНИЕ-----
CALL TOCHKA
;-----ЕСЛИ ЦЕЛОЕ<>0, ТО ","- НЕ НУЖНА-----
SUP@:
JMP NEAR:EXIT
;-----
NET@:
;ВСЕ ОСТАЛЬНЫЕ СЛУЧАИ - ЕСЛИ SI<>LEA S2 И AL<>0
;ВЫВОД ПЕРВОЙ ЦИФРЫ, ЗАПЯТОЙ, ВТОРОЙ ЦИФРЫ
CALL CHET_NECHET
JE NECHETNOE_@
;ЕСЛИ SI ЕСТЬ НЕЧЕТНОЕ ЧИСЛО
CHETNOE_@:
CALL SDVIG_RAZRAD_DESATOK
JE ES_DEC_RABNO_NOL_@
;-----ЕСЛИ РАЗРЯД 10-К НЕ РАВЕН НУЛЮ-----
CALL PECHAT_CHETNOE_DEC
CMP DL,0H
JNE SUP@@
CALL TOCHKA
;-----ЕСЛИ ЦЕЛОЕ<>0, ТО ","- НЕ НУЖНА-----
SUP@@:CALL PECHAT_CHETNOE_ED
JMP NET_ES_DEC_RABNO_NOL_@
;-----
ES_DEC_RABNO_NOL_@:
CALL PECHAT_CHETNOE_ED
CMP DL,0H
JNE SUP@@@
CALL TOCHKA
SUP@@@:
;ЕСЛИ ЦЕЛОЕ<>0, ТО ","- НЕ НУЖНА
NET_ES_DEC_RABNO_NOL_@:NOP
JMP NOU_NECHETNOE_@
;-----
NECHETNOE_@:CALL PECHAT_NECHETNOE
CMP DL,0H
JNE SUP@@@
CALL TOCHKA
SUP@@@:
;ЕСЛИ ЦЕЛОЕ<>0, ТО ","- НЕ НУЖНА
;-----
NOU_NECHETNOE_@:
;ПЕЧАТЬ ОСТАЛЬНОЙ ЧАСТИ ДРОБНОГО ЧИСЛА
DEC SI
MOV AL,[SI]
CALL CHET_NECHET
JE NECHETNOE_@@
;ФЛАГ УСТАНАВЛИВАЕТСЯ ЕСЛИ ЧИСЛО НЕЧЕТНО
CHETNOE_@@:CALL PECHAT_CHETNOE_DEC
CALL PECHAT_CHETNOE_ED
JMP NOU_NECHETNOE_@@
NECHETNOE_@@:CALL PECHAT_NECHETNOE
NOU_NECHETNOE_@@:CMP SI,12
JE EXIT
JMP NOU_NECHETNOE_@
;-----
;-----Выход из программы-----
;-----
EXIT:
CALL EXP ;ДОПОЛНИТЕЛЬНАЯ ЧАСТЬ - ЭКСПОНЕНТА
CMP DL,1
JE OK_PLUS ;ЕСЛИ ЦЕЛАЯ ЧАСТЬ<>0
;-----
CALL OPREDELENIE_K3_DROB
CALL NOL@@
MOV CL,K3
CALL ZNAKMIN
JMP METKAPORADOK
;-----

```

```

OK_PLUS:
CALL ZNAKPL
METKAPORADOK:CALL PORADOK
CALL NOL@@
;-----
MOV AX,1000H
INT 16H
EXIT@:
MOV AX,4F02H ;ВОССТАНОВИЛИ ПРЕЖНИЙ ВИДЕОРЕЖИМ
XOR BH,BH
MOV BL,VIDEO
INT 10H
MOV AX,4C00H
INT 21H
;-----
;-----ГЛАВНЫЕ ПРОЦЕДУРЫ ПРОГРАММЫ-----
;-----
NOL@@ PROC
    PUSH SI
    PUSH CX
    LEA SI,S2
    ADD SI,7
    MOV CX,8
    CIKL_NOL_OBE_CHASTI:
        MOV AL,[SI]
        CMP AL,00H
        JNE NET_NOL_OBE_CHASTI
        DEC SI
        LOOP CIKL_NOL_OBE_CHASTI
    MOV K3,0
    NET_NOL_OBE_CHASTI:
    POP CX
    POP SI
    RET
NOL@@ ENDP
;-----
OPREDELENIE_K3_DROB PROC
    PUSH SI
    PUSH AX
    LEA SI,S2
    ADD SI,7
    ;-----
    CIKL_K3:
        MOV AL,[SI]
        CMP AL,00H
        JNE NO_CIKL_K3
        CALL CHET_NECHET
        JE MET
        INC K3
        MET:INC K3
        DEC SI
        JMP CIKL_K3
    ;-----
    NO_CIKL_K3:CALL CHET_NECHET
        JE FLOPY
        ;ЕСЛИ ЧЕТНАЯ ПОЗИЦИЯ
        INC K3
        SHR AL,4
        CMP AL,00
        JNE NO_MET@
        INC K3
        JMP NO_MET@
        FLOPY:INC K3
    NO_MET@:
    POP AX
    POP SI
    RET
OPREDELENIE_K3_DROB ENDP
;-----

```

```

SDVIG_RAZRAD_DESATOK PROC
    PUSH AX
    SHR AL, 4 ;ПРОВЕРИМ РАЗРЯД 10-К - ОН ДОЛЖЕН<>0
    CMP AL, 00H ;ЕСЛИ ПРИ СДВИГЕ РАЗРЯД 10-К=0, ТО...
    POP AX
    RET
SDVIG_RAZRAD_DESATOK ENDP
;-----
PECHAT_NECHEТNOE PROC
    PUSH AX
    CALL HEX_OR_DEC
    MOV AH, 7
    MOV ES:[BP], AX
    ADD BP, 2
    POP AX
    RET
PECHAT_NECHEТNOE ENDP
;-----
ТОЧКА PROC
    PUSH AX
    MOV AL, 44
    MOV AH, 7
    MOV ES:[BP], AX
    ADD BP, 2
    POP AX
    RET
ТОЧКА ENDP
;-----
PECHAT_CHEТNOE_DEC PROC
    PUSH AX
    SHR AL, 4 ;СДВИГАЕМ 2Х РАЗРЯДНОЕ ЧИСЛО ВЛЕВО НА 4 БИТА
    CALL HEX_OR_DEC
    MOV AH, 7
    MOV ES:[BP], AX
    ADD BP, 2
    POP AX
    RET
PECHAT_CHEТNOE_DEC ENDP
;-----
PECHAT_CHEТNOE_ED PROC
    PUSH AX
    SHL AL, 4 ;ОЧИЩАЕМ РАЗРЯД ДЕСЯТОК В ДВУХРАЗРЯДНОМ ЧИСЛЕ
    SHR AL, 4
    CALL HEX_OR_DEC
    MOV AH, 7
    MOV ES:[BP], AX
    ADD BP, 2
    POP AX
    RET
PECHAT_CHEТNOE_ED ENDP
;-----
PECHAT_NOL PROC
    PUSH AX
    MOV AL, 30H
    MOV AH, 7
    MOV ES:[BP], AX
    ADD BP, 2
    POP AX
    RET
PECHAT_NOL ENDP
;-----ОПРЕДЕЛЯЕТ КАКОЕ ЧИСЛО - ДЕСЯТИЧНОЕ ИЛИ ШЕСТНАДЦАТЕРИЧНОЕ-----
HEX_OR_DEC PROC
    CMP AL, 9
    JG HEX ;ЕСЛИ ЧИСЛО ШЕСТНАДЦАТЕРИЧНОЕ
    OR AL, BL ;ПРОСТО ДОПОЛНЯЕМ ДО ASCII КОДА
    JMP NET_HEX
HEX:ADD AL, 37H ;ПРОСТО ДОПОЛНЯЕМ ДО ASCII КОДА
NET_HEX:RET
HEX_OR_DEC ENDP

```

```

;-----Определяет четность позиций-----
CHET_NECHET PROC
    PUSH SI
    CIKL:
        CMP SI,1
        JLE NO_CIKL
        SUB SI,2
        JMP CIKL
    NO_CIKL: CMP SI,1
    POP SI
    RET
CHET_NECHET ENDP

;-----ПЕЧАЕТ БУКВУ ЭКСПОНЕНТЫ-----
EXP PROC
    MOV AH,7
    MOV AL,80
    MOV ES:[BP],AX
    ADD BP,2
    RET
EXP ENDP

;-----ПЕЧАТАЕТ ЗНАК ЭКСПОНЕНТЫ "+"-----
ZNAKPL PROC
    MOV AH,7
    MOV AL,43
    MOV ES:[BP],AX
    ADD BP,2
    RET
ZNAKPL ENDP

;-----ПЕЧАТАЕТ ЗНАК ЭКСПОНЕНТЫ "-"-----
ZNAKMIN PROC
    MOV AH,7
    MOV AL,45
    MOV ES:[BP],AX
    ADD BP,2
    RET
ZNAKMIN ENDP

;-----ПЕЧАТАЕТ ПОРЯДОК ЭКСПОНЕНТЫ-----
PORADOK PROC
    MOV AH,7
    MOV AL,K3
    OR AL,30h
    MOV ES:[BP],AX
    ADD BP,2
    RET
PORADOK ENDP

;-----
;-----ПРОЦЕДУРА ВЫВОДА ЦИФРЫ НА ЭКРАН-----
;-----
CIFRA PROC
    MOV AH,7
    MOV ES:[BP],AX
    ADD BP,2
    RET
CIFRA ENDP

;-----
;-----Процедура перевода числа из одной системы в другую-----
;-----
PEREBOD PROC
; **Работа с регистром AL- сдвиг старшего полубайта на 1 к младшему полубайту**
SHAG_1:
    MOV BL,AL
    SHL BL,4
    SHR BL,4
    SHR AL,4
    SHL AL,4
    SHR AL,1
    OR AL,BL
; Работа с регистром AX, а именно с младшим полубайтом AH и старшим полубайтом
; AL - сдвигаем влево на 2 позиции

```

```
SHAG_2:
    MOV BX,AX
    SHL BH,4
    SHR BH,2
    MOV DH,BH
    SHL BH,4
    OR AL,BH
    MOV BH,DH
    SHR BH,4
    SHR AH,4
    SHL AH,4
    OR AH,BH
; **Работа с регистром AH– сдвиг старшего полубайта на 3 к младшему полубайту**
SHAG_3:
    MOV BH,AH
    SHL AH,4
    SHR AH,4
    SHR BH,4
    SHL BH,4
    SHR BH,3
    OR AH,BH
    RET
PEREBOD ENDP
END MAIN
```