Задание 1

УСЛОВИЕ ЗАДАНИЯ:

Ввести в ячейки 120,123,... числа 7,B,...,10F (все данные в 16-ой системе)

ПЛАН РЕШЕНИЯ:

1. Задание начальных значений
   (начального числа= 7,
    стартового адреса= 120,
    числа шагов цикла= 43).

Примечание. Расчет значения счетчика цикла:

CX= (10F - 7):4 + 1 = 108:4 + 1 = 42 + 1 = 43

```
 10F 108¦ 4              16-ая             10-ая
-  7   10 ¦--          1)       10:4 = 4   16:4 = 4
 108 --- 42           2)        8:4 = 2    8:4 = 2
        08
        08
        --
         0
```

2. Организация цикла: [DI] <- AL
   (последовательное заполнение ячеек 120,123,126,... числами 7,B,F,...,10F)

3. Завершение работы программы (INT 20)

КОД ПРОГРАММЫ:

```
100 MOV AL,07      ; AL  <- 7 (начальное число = 7)
102 MOV DI,0120    ; DI  <- 120      (стартовый адрес = 120)
105 MOV CX,0043  ; CX  <- 43       (число шагов цикла= 43)

            ; цикл по перебору ячеек диапазона:
108 MOV [DI],AL    ;[DI] <- AL       (размещение в очередной ячейке памяти очередного числа)
10A ADD AL,04      ; AL  <- AL + 4 (наращивание числа: +4)
10C ADD DI,0003    ; DI  <- DI + 3   (наращивание адреса: +3)
10F LOOP 108       ; CX  <- CX - 1 После чего, если CX <> 0, следует переход по адресу 108 (команда цикла)

111 INT 20       ; завершение работы
```

ПРОВЕРКА РЕШЕНИЯ:

Для проверки решения уменьшим значение счетчика цикла CX до 3.

1-ый шаг: AL = 7
2-ой шаг: DI = 120
3-ий шаг: CX = 3

4-ый шаг: [120] = 7
5-ый шаг: AL = 7 + 4 = B
6-ой шаг: DI = 120 + 3 = 123
7-ой шаг: CX = 3 - 1 = 2 (2 > 0, поэтому следует переход на адрес 108)
8-ой шаг: [123] = B
9-ый шаг: AL = B + 4 = F
10-ый шаг: DI = 123 + 3 = 126
11-ый шаг: CX = 2 - 1 = 1 (1 > 0, поэтому следует переход на адрес 108)
12-ый шаг: [126] = F
13-ый шаг: AL = F + 4 = 13
14-ый шаг: DI = 126 + 3 = 129
15-ый шаг: CX = 1 - 1 = 0 (0 = 0, цикл завершен)

16-ый шаг: INT 20 - завершение работы.

РЕЗУЛЬТАТ РЕШЕНИЯ:

[120] = 7
[123] = B
[126] = F

Задание 2

УСЛОВИЕ ЗАДАНИЯ:

Ввести в ячейки 120,123,... числа 7,B,...,10F (все данные в 16-ой системе)

ПЛАН РЕШЕНИЯ:

1. Задание начальных значений до начала цикла
  (стартового адреса= 120,
   ASCII-кода символа "5"= 35,
   ASCII-кода символа "9"= 39,
   числа шагов цикла= 50).

Примечание. Расчет значения счетчика цикла:

CX= (16F - 120) + 1 = 4F + 1 = 50

2. Организация цикла: [DI] <-> AH

(последовательная проверка ячеек с целью поиска в них числа 39,
заполнение очередной ячейки кодом 35, если найден символ "9",
наращивание содержимого регистра DI)

3. Завершение работы программы (INT 20)

КОД ПРОГРАММЫ:

```
100 MOV DI,0120   ; DI   <- 120
103 MOV AL,35     ; AL   <- 35
105 MOV AH,39     ; AH   <- 39
107 MOV CX,0050   ; CX   <- 50

            ; цикл по перебору ячеек диапазона:
10A CMP [DI],AH    ;[DI] <-> AH
10C JNZ 0110 ; переход, если имеет место "Not Zero"
10E MOV [DI],AL    ; [DI] <- AL    (AL = 35)
110 INC DI      ; DI   <- DI + 1
111 LOOP 010A     ; CX   <- CX - 1 После чего, если CX <> 0, следует переход по адресу 10A

113 INT 20      ; завершение работы
```

ПРОВЕРКА РЕШЕНИЯ:

Для проверки решения уменьшим значение счетчика цикла CX до 3.
В ячейках 120-122 разместим строку "193"

1-ый шаг: DI = 120
2-ой шаг: AL = 35
3-ий шаг: AH = 39
4-ый шаг: CX = 3

5-ый шаг: [120]= 31 <-> 39 Не равно, т.е. имеет место "Not Zero"
6-ой шаг: переход на адрес 110, т.к. имеет место "Not Zero"
7-ой шаг: DI = 120 + 1 = 121
8-ой шаг: CX = 3 - 1 = 2 (2 > 0, поэтому следует переход на адрес 10A)
9-ый шаг: [121]= 39 <-> 39 Равно, т.е. имеет место "Zero"
10-ый шаг: переход на адрес 110 не следует, т.к. имеет место "Zero"
11-ый шаг: [121]= 35
12-ый шаг: DI = 121 + 1 = 122
13-ый шаг: CX = 2 - 1 = 1 (1 > 0, поэтому следует переход на адрес 10A)
14-ый шаг: [122]= 33 <-> 39 Не равно, т.е. имеет место "Not Zero"
15-ый шаг: переход на адрес 110, т.к. имеет место "Not Zero"
16-ый шаг: DI = 122 + 1 = 123
17-ый шаг: CX = 1 - 1 = 0 (0 = 0, цикл завершен)

18-ый шаг: INT 20 - завершение работы.

РЕЗУЛЬТАТ РЕШЕНИЯ:

В ячейках 120-122 находятся символы "153"

Задание 3

УСЛОВИЕ ЗАДАНИЯ:

Подсчитать количество символов-цифр в диапазоне ячеек 140..17F. Результат сохранить в ячейке 122.

ПЛАН РЕШЕНИЯ:

1. Задание начальных значений
   (начального значения счетчика символов-цифр= 0,
   стартового адреса= 140,
   числа шагов цикла= 40
   ASCII-кода минимальной цифры= 30,
   ASCII-кода максимальной цифры= 39,
).

2. Организация цикла: ([BP] <-> DL) и ([BP] <-> DH)
   (последовательный перебор ячеек 140..17F
   с целью выделения символов-цифр,
   ADCII-коды которых заключены в пределах от 30 до 39
   (в этом случае AL увеличивается на 1)
)

3. Сохранение числа найденных символов-цифр (в ячейке памяти 0122).

4. Завершение работы программы (INT 20)

КОД ПРОГРАММЫ:

```
100  MOV  AL,00    ; в регистре AL накапливается число символов-цифр
102  MOV  BP,0140       ; в регистр BP заносится стартовый адрес ячеек диапазона
105  MOV  CX,0040       ; в регистр CX заносится количество ячеек диапазона
108  MOV  DL,30    ; в регистр DL заносится  ADCII-код символа "0"
10A  MOV  DH,39    ; в регистр DH заносится  ADCII-код символа "9"

                   ; цикл по перебору ячеек диапазона:
10C  CMP  [BP],DL  ; сравнение содержимого очередной ячейки с цифрой "0"
10F  JB   117   ; переход, если "Below/меньше"(отсечение символов < "0", не цифр)
111  CMP  [BP],DH ; сравнение содержимого ячейки с цифрой "9"
114  JA   117   ; переход, если "Above/больше"(отсечение символов > "9", не цифр)
116  INC  AL    ; наращивание значения счетчика символов-цифр AL, если не было переходов "Below" и "Above"
117  INC  BP    ; наращивание значения указателя адресов BP
118  LOOP 010C     ; CX <- CX - 1 После чего, если CX <> 0, следует переход по адресу 10C

11A  MOV  BP,0122       ; занесение в регистр BP адреса ячейки памяти 0122
11D  MOV  [BP],AL  ; сохранение в ячейке памяти 0122 значения счетчика символов-цифр

11F  INT  20    ; завершение работы
```

Задание 4

УСЛОВИЕ ЗАДАНИЯ:

В ячейке памяти 140 находится число, не превышающее 63.
Вывести это число на экран монитора (в десятичной системе счисления).

КОД ПРОГРАММЫ:

```
100  MOV  BP,0140        ; загрузка в регистр BP адреса 140
103  MOV  AL,[BP+00]     ; загрузка в регистр AL содержимого ячейки 0140
106  MOV  AH,0A    ; загрузка в регистр AH числа 10 (основание 10-ой системы)
108  MOV  DX,0000        ; обнуление регистра DX

                   ; цикл по нахождению числа десятков и единиц:
10B  CMP  AL,AH    ; сравнение содержимого ячейки 0140 с числом 10
10D  JB   0115 ; переход, если имеет место "Below/меньше" (в числе- одна цифра)
10F  SUB  AL,AH    ; вычитание с целью определения цифры в разряде десятков
111  INC  DH   ; наращивание числа десятков
113  JMP  010B      ; переход по адресу 010B (цикл по определению числа десятков и числа единиц)

115  MOV  DL,AL    ; загрузка в регистр DL числа единиц исходного числа
117  ADD  DL,30    ; превращение числа единиц в соответствующий символ
```

```
11A  ADD  DH,30    ; превращение числа десятков в соответствующий символ

11D  PUSH DX       ; сохранение символов единиц и десятков в стеке

11E  MOV  DL,DH    ; загрузка в регистр DL числа десятков исходного числа
120  MOV  AH,02    ; загрузка в регистр AH номера функции прерывания INT  21
122  INT  21     ; вывод символа (числа десятков) на экран (в DL - ASCII-код символа)

124  POP  DX  ; извлечение из стека в регистр DX числа десятков и единиц
125  MOV  AH,02    ; подготовка к выводу на экран числа единиц (регистр DL)
127  INT  21     ; вывод символа (числа единиц) на экран (в DL - ASCII-код символа)

129  INT  20     ; завершение работы
```

Задание 5

УСЛОВИЕ ЗАДАНИЯ:

Определить разницу между максимальным и минимальным числами диапазона 140..16F.
Результат сохранить в ячейке 13F.

КОД ПРОГРАММЫ:

```
100  MOV  BP,0140        ; загрузка в регистр BP начального адреса заданного диапазона
103  MOV  CX,0030        ; в регистр CX заносится количество ячеек диапазона
106  MOV  DL,FF    ; в регистр DL заносится число FF (начальное минимальное число)
108  MOV  DH,00    ; в регистр DH заносится число 0 (начальное максимальное число)

                   ; цикл по нахождению минимального и максимального значений:
10A  MOV  AL,[BP+00]    ; в регистр AL заносится содержимое очередной ячейки диапазона
10D  CMP  AL,DL     ; содержимое ячейки сравнивается с минимальным числом
10F  JA   0113 ; если имеет место "Above/больше", минимальное число не меняется
```

```
111  MOV  DL,AL    ; обновление минимального числа (текущее число оказалось меньше)
113  CMP  AL,DH    ; содержимое ячейки сравнивается с максимальным числом
115  JB   0119 ; если имеет место "Below/меньше", максимальное число не меняется
117  MOV  DH,AL    ; обновление максимального числа (текущее число оказалось больше)
119  INC  BP    ; наращивание адреса текущей ячейки памяти
11A  LOOP 010A     ; последняя строка цикла по перебору ячеек памяти

11C  SUB  DH,DL    ; определение разности (максимальное минус минимальное)
11E  MOV  [013F],DH      ; сохранение в ячейке 13F найденной разности

122  INT  20    ; завершение работы
```

Задание 6

УСЛОВИЕ ЗАДАНИЯ:

Ввести с клавиатуры строку.
Определить, сколько цифр содержит введенная строка.
Результат вывести на экран.

КОД ПРОГРАММЫ:

```
100  JMP  130 ; перепрыгивание через буфер данных

102  DB   'Wait string entrance...', 0D, 0A, '$'
11C  DB   0D, 0A, 0A, 'Cifra amount:  $'

130  MOV  DX,0102        ; задание адреса строки 'Wait string entrance...'
133  MOV  AH,09    ; задание в AH номера функции (вывод строки на экран)
135  INT  21    ; вывод строки на экран (102 - адрес выводимой строки)

137  MOV  BP,0188        ; задание в BP адреса буфера клавиатуры (адрес= 188)
13A  MOV  AL,0D    ; 0D - ограничение на число вводимых с клавиатуры символов
13C  MOV  [BP+00],AL    ; загрузка в ячейку [188] числа-ограничителя ( 0D )

13F  MOV  DX,BP   ; задание в DX адреса буфера клавиатуры
141  MOV  AH,0A   ; задание в AH номера функции ( 0A - ввод строки)
143  INT  21    ; ввод строки (188 - адрес буфера строки)
```

```
145  MOV  AL,00    ; начальное значение счетчика числа цифр (AL= 0)
147  MOV  CH,00    ; CH= 0
149  MOV  CL,[BP+01]   ; загрузка в CL числа символов введенной строки ([189])
14C  MOV  DL,30    ; ASCII-код (30) символа "0"
14E  MOV  DH,39    ; ASCII-код (39) символа "9"

                   ; цикл по выявлению цифр введенной строки:
150  CMP  [BP+02],DL    ; сравнение очередного символа строки с символом "0"
153  JB   15C  ; если "Below/меньше" переход на 15C (текущий символ - не цифра)
155  CMP  [BP+02],DH   ; сравнение очередного символа строки с символом "9"
158  JA   15C  ; если "Above/больше" переход на 15C (текущий символ - не цифра)
15A  INC  AL   ; наращивание значения AL - счетчика числа цифр
15C  INC  BP   ; наращивание адреса BP
15D  LOOP 150       ; CX <- CX - 1 Если CX <> 0, переход на адрес 150

                   ; обработка цифр числа в AL (возможные значения: от 0 до 12):
15F  CMP  AL,09    ; AL <-> 09
161  JBE  16B ; переход, если "Below или Equal /меньше или равно" (число < 10, т.е. состоит из одной цифры)
                   ; обработка чисел (значения: от 10 до 12):
163  SUB  AL,0A    ; определение числа единиц в анализируемом числе
165  MOV  AH,31    ; задание числа десятков в AH в виде символа "1" (один десяток)
167  ADD  AL,30    ; преобразование числа единиц в символ (AL <- AL + 30)
169  JMP  16F ; переход по адресу 16F

16B  ADD  AL,30    ; преобразование единиц в символ (для чисел: от 0 до 9)
16D  MOV  AH,30    ; занесение в AH символа "0"

16F  PUSH AX       ; сохранение в стеке 1-го и 2-го символов ответа

170  MOV  DX,011C       ; задание адреса строки 'Cifra amount:'
173  MOV  AH,09    ; задание в AH номера функции (вывод строки на экран)
175  INT  21    ; вывод строки на экран (11C - адрес выводимой строки)

177  POP  AX  ; восстановление в AX 1-го и 2-го символов ответа
178  PUSH AX       ; сохранение в стеке 1-го и 2-го символов ответа

179  MOV  DL,AH    ; занесение в DL 1-го символа ответа
17B  MOV  AH,02    ; занесение в AH номера функции (02 - вывод символа на экран)
17D  INT  21    ; вывод символа на экран (DL - ASCII-код символа)

17F  POP  AX  ; восстановление в AX 1-го и 2-го символов ответа

180  MOV  DL,AL    ; занесение в DL 2-го символа ответа
182  MOV  AH,02    ; занесение в AH номера функции
184  INT  21    ; вывод символа (DL - ASCII-код символа)

186  INT  20    ; завершение работы
```

Задание 7

УСЛОВИЕ ЗАДАНИЯ:

Составить исходный модуль, удовлетворяющий следующим условиям:
Вывести на экран сумму ряда однозначных чисел,
заданных в 10-ой системе счисления.
Количество чисел не превышает 3 штук.
Числа находятся в буфере и готовы к выводу.

КОД ИСХОДНОГО МОДУЛЯ:

```
model      small
_GROUP GROUP code_segm, data_segm
ASSUME cs: _GROUP, ds: _GROUP
code_segm      SEGMENT
               ORG        100h
start:
        mov dx,   _GROUP
        mov ds,   dx

        ; строки, предшествующие данной - стандартные
        ; строки, следующие за данной - нуждаются в конкретных комментариях

        mov bx,   OFFSET Numbers  ; загрузка в bx адреса буфера "Numbers"

        mov ax,   0       ; обнуление регистра, в котором будет размещена сумма

        ; цикл для нахождения суммы чисел, содержащихся в ячейках буфера:
```

```
m_1:    mov dl,   [bx]  ; загрузка в dl содержимого очередной ячейки буфера
        cmp dl,   255  ; сравнение содержимого ячейки с числом 255
        jz    m_2      ; завершение цикла, если содержимое ячейки = 255
        add  al,  dl   ; добавление к сумме (al) содержимого очередной ячейки
        adc  ah,  0    ; учет возможного переноса из 0-го в 1-ый разряд
        inc   bx       ; наращивание адреса буфера "Numbers"
        jmp  m_1       ; замыкание цикла (суммирование содержимого ячеек буфера)

m_2:    mov cl,   10   ; подготовка к выполнению деления на 10
        div   cl       ; деление AX на 10 с целью определения цифр числа (al- результат деления,  ah- остаток)
        push      ax        ; сохранение AX в стеке
        mov dl,   al   ; занесение в dl старшей цифры суммы
        add  dl,  30h  ; преобразование числа-цифры в символ
        mov ah,   2    ; номер функции, обеспечивающей вывод на экран символа
        int   21h      ; вывод на экран старшей цифры числа

        pop  ax        ; восстановление в AX 1-го и 2-го символов суммы

        mov dl,   ah   ; занесение в dl младшей цифры суммы
        add  dl,  30h  ; преобразование младшей цифры суммы в соответствующий символ
        mov ah,   2
        int   21h      ; вывод на экран младшей цифры числа

        mov ah,   4ch  ; номер функции, обеспечивающей завершение работы
        int   21h      ; завершение работы
code_segm    ENDS

data_segm    SEGMENT
Numbers DB 10 DUP(7,8,6,255)    ; буфер "Numbers" (три числа и метка "255" - знак конца буфера)
data_segm    ENDS
        END      start
```

ПРОВЕРКА РЕШЕНИЯ:

Проверку начинаем с команды "mov bx, OFFSET Numbers".
Предположим, что буфер "Numbers" - совокупность ячеек с адресами 160..163

1-ый шаг: bx = 160
2-ой шаг: ax = 0

3-ий шаг: dl= [160]= 7

4-ый шаг: 7 <-> 255 Не равно, т.е. имеет место "Not Zero"
5-ый шаг: перехода нет, т.к. имеет место "Not Zero"
6-ой шаг: al = 0 + 7 = 7
7-ой шаг: ah = 0 + 0 + 0 = 0
8-ой шаг: bx = 160 + 1 = 161
9-ый шаг: переход на метку m_1
10-ый шаг: dl= [161]= 8
11-ый шаг: 8 <-> 255 Не равно, т.е. имеет место "Not Zero"
12-ый шаг: перехода нет, т.к. имеет место "Not Zero"
13-ый шаг: al = 7 + 8 = 15
14-ый шаг: ah = 0 + 0 + 0 = 0
15-ый шаг: bx = 161 + 1 = 162
16-ый шаг: переход на метку m_1
17-ый шаг: dl= [162]= 6
18-ый шаг: 6 <-> 255 Не равно, т.е. имеет место "Not Zero"
19-ый шаг: перехода нет, т.к. имеет место "Not Zero"
20-ый шаг: al = 15 + 6 = 21
21-ый шаг: ah = 0 + 0 + 0 = 0
22-ой шаг: bx = 162 + 1 = 163
23-ий шаг: переход на метку m_1
24-ый шаг: dl= [163]= 255
25-ый шаг: 255 <-> 255 Равно, т.е. имеет место "Zero"
26-ой шаг: т.к. имеет место "Zero", следует переход на метку m_2

27-ой шаг: cl= 10
28-ой шаг: ax : cl = 21 : 10 = 2(1), где al=2 - результат деления, ah=1 - остаток
29-ый шаг: сохранение в стеке числа ax= 0102 hex
30-ый шаг: dl = 2
31-ый шаг: dl = 2 + 30h = 32h
32-ой шаг: ah = 2
33-ий шаг: вывод на экран символа "2"
34-ый шаг: ax= 0102 hex
35-ый шаг: dl = 1
36-ой шаг: dl = 1 + 30h = 31h
37-ой шаг: ah = 2
38-ой шаг: вывод на экран символа "1"

39-ый шаг: ah = 4ch (номер функции, обеспечивающей завершение работы)
40-ой шаг: int 21h - завершение работы.

РЕЗУЛЬТАТ РЕШЕНИЯ:

На экран выведены символы "21"

Задание 8

УСЛОВИЕ ЗАДАНИЯ:

Составить исходный модуль, удовлетворяющий следующим условиям:
Вывести на экран сумму содержимого ряда ячеек памяти,
представленную в 10-ой системе счисления.
Количество ячеек памяти не превышает 25 штук.

КОД ИСХОДНОГО МОДУЛЯ:

```
model      small
_GROUP GROUP code_segm, data_segm
ASSUME cs: _GROUP, ds: _GROUP
code_segm     SEGMENT
              ORG       100h
start:
        mov dx,   _GROUP
        mov ds,   dx
        mov bp,   OFFSET Numbers  ; загрузка в bp адреса буфера "Numbers"
```

```
        mov ax,   0     ; обнуление регистра, в котором будет размещена сумма

        ; цикл для нахождения суммы чисел, содержащихся в ячейках буфера:
m_1:    mov dl,   [bp]
        cmp dl,   255
        jz    m_2       ; завершение цикла, если содержимое ячейки = 255
        add al,   dl    ; сумма накапливается в регистре  ax
        adc ah,   0
        inc   bp
        jmp   m_1

m_2:    mov dx,   0FFFFh   ; символы цифр ответа будут накапливаться в стеке
                           ; для этого в стеке необходимо разместить метку
        push      dx       ; занесение в стек метки  "0FFFFh"
        mov bx,   10   ;
        ; цикл для накопления в стеке цифр найденной суммы:
m_3:    cmp ax,   bx   ;
        jb    m_4      ; переход, если в сумме - всего одна цифра
                       ; в сумме - 2 или 3 цифры:
        mov dx,   0    ;
        div   bx       ; (dx:ax) : bx = ax (dx) (ax- результат деления,  dx- остаток)
        push      dx       ; сохранение в стеке остатка деления на 10 (т.е. младшей из цифр)
        jmp   m_3      ; цикл, пока не останется единственная из цифр числа - старшая цифра

m_4:    push      ax       ; в ax - старшая цифра, сохраняем ее в стеке
        ; цикл для вывода на экран всех цифр числа, накопленных в стеке:
m_5:    pop   ax       ; выталкиваем из стека в ax очередную цифру
        cmp al,   FFh  ; проверка на предмет окончания цифр в стеке (метка FFh)
        jz    m_6      ; завершение вывода цифр числа из стека
        mov dl,   al   ; занесение в dl очередной цифры
        add dl,   30h  ; превращение очередной цифры в символ
        mov ah,   2
        int   21h      ; вывод на экран очередной цифры-символа накопленной суммы
        jmp   m_5      ; зацикливание процедуры вывода цифр из стека

m_6:    mov ah,   4ch
        int   21h
code_segm   ENDS

data_segm   SEGMENT
Numbers DB 26 DUP(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,255)
data_segm   ENDS
        END     start
```

ПРОВЕРКА РЕШЕНИЯ:

Проверку начинаем с команды "mov bx, OFFSET Numbers".
Предположим, что буфер "Numbers" - совокупность ячеек с адресами 160..163

1-ый шаг: bx = 160
2-ой шаг: ax = 0

3-ий шаг: dl= [160]= 7
4-ый шаг: 7 <-> 255 Не равно, т.е. имеет место "Not Zero"
5-ый шаг: перехода нет, т.к. имеет место "Not Zero"
6-ой шаг: al = 0 + 7 = 7
7-ой шаг: ah = 0 + 0 + 0 = 0
8-ой шаг: bx = 160 + 1 = 161
9-ый шаг: переход на метку m_1
10-ый шаг: dl= [161]= 8
11-ый шаг: 8 <-> 255 Не равно, т.е. имеет место "Not Zero"
12-ый шаг: перехода нет, т.к. имеет место "Not Zero"
13-ый шаг: al = 7 + 8 = 15
14-ый шаг: ah = 0 + 0 + 0 = 0
15-ый шаг: bx = 161 + 1 = 162
16-ый шаг: переход на метку m_1
17-ый шаг: dl= [162]= 6
18-ый шаг: 6 <-> 255 Не равно, т.е. имеет место "Not Zero"
19-ый шаг: перехода нет, т.к. имеет место "Not Zero"
20-ый шаг: al = 15 + 6 = 21
21-ый шаг: ah = 0 + 0 + 0 = 0
22-ой шаг: bx = 162 + 1 = 163
23-ий шаг: переход на метку m_1
24-ый шаг: dl= [163]= 255
25-ый шаг: 255 <-> 255 Равно, т.е. имеет место "Zero"
26-ой шаг: т.к. имеет место "Zero", следует переход на метку m_2

27-ой шаг: cl= 10
28-ой шаг: ax : cl = 21 : 10 = 2(1), где al=2 - результат деления, ah=1 - остаток
29-ый шаг: сохранение в стеке числа ax= 0102 hex
30-ый шаг: dl = 2
31-ый шаг: dl = 2 + 30h = 32h
32-ой шаг: ah = 2

33-ий шаг: вывод на экран символа "2"
34-ый шаг: ax= 0102 hex
35-ый шаг: dl = 1
36-ой шаг: dl = 1 + 30h = 31h
37-ой шаг: ah = 2
38-ой шаг: вывод на экран символа "1"

39-ый шаг: ah = 4ch (номер функции, обеспечивающей завершение работы)
40-ой шаг: int 21h - завершение работы.

РЕЗУЛЬТАТ РЕШЕНИЯ:

На экран выведены символы "21"

Задание 9

УСЛОВИЕ ЗАДАНИЯ:

Составить исходный модуль, удовлетворяющий следующим условиям:
Вывести на экран (в 10-ой системе счисления) ряд чисел,
не превышающих числа 254.
Числа находятся в буфере и готовы к выводу.

КОД ИСХОДНОГО МОДУЛЯ:

```
model      small
_GROUP GROUP code_segm, data_segm
ASSUME cs: _GROUP, ds: _GROUP
code_segm     SEGMENT
              ORG       100h
start:
       mov dx,   _GROUP
       mov ds,   dx

       mov bp,   OFFSET Number    ; загрузка в bp адреса буфера "Number"
       mov bx,   10

       ; цикл для вывода на экран чисел, содержащихся в ячейках буфера (1, 2 или 3 цифры в числе):
m_1:       mov dx,   0FFFFh
       push       dx          ; метка "0FFFFh" в стеке (обозначает начало цифр числа)
       mov ah,   0
       mov al,   [bp]  ; считывание содержимого очередной ячейки буфера
       cmp al,   FFh
```

```
      jz    m_0       ; вывод чисел завершен (переход на метку, обозначающую конец программы)

m_2:  cmp ax,  bx
      jb    m_3       ; переход, если в числе - всего одна цифра
                      ; в числе - 2 или 3 цифры:
      mov dx,  0
      div   bx        ; (dx:ax) : bx = ax (dx) (ax- результат деления,  dx- остаток)
      push      dx          ; сохранение в стеке остатка деления на 10 (т.е. младшей из цифр)
      jmp   m_2       ; цикл, пока не останется единственная из цифр числа - старшая цифра

m_3:  push      ax          ; в ax - старшая цифра, сохраняем ее в стеке

      ; цикл для вывода на экран всех цифр числа, накопленных в стеке:
m_4:  pop   ax          ; выталкивание из стека в ax очередной цифры числа
      cmp al,   FFh ; проверка на предмет окончания цифр в стеке (метка FFh)
      jz    m_5       ; переход на m_5 (подготовка к выводу следующего числа)
      mov dl,   al    ; занесение в dl очередной цифры
      add   dl,   30h ; превращение очередной цифры в символ
      mov ah,   2     ;
      int   21h       ; вывод на экран очередной цифры-символа числа
      jmp   m_4       ; зацикливание процедуры вывода цифр из стека

m_5:  mov dl,   20h  ; занесение в dl ASCII-кода пробела
      mov ah,   2
      int   21h       ; вывод на экран пробела (для разделения соседних чисел)

      inc   bp        ; наращивание bp (для перехода к следующей ячейке буфера)
      jmp   m_1       ; зацикливание процесса перебора чисел буфера
m_0:
      mov ah,   4ch
      int   21h
code_segm     ENDS

data_segm     SEGMENT
Number   DB 6 DUP(7,60,170,202,254,255)
data_segm     ENDS
      END       start
```