

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

Лабораторная работа № 1

по Ассемблеру на тему:

"Разработка машинно-ориентированных программ обработки текстов"

Вариант № 3

*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.....	3

Отчет о выполненной работе

1. Текст постановки задачи.

Программа вводит с терминала ключевое слово. Затем вводится строка слов, разделенных одним или более пробелов. Выполняется поиск ключевого слова во введенном списке слов. Результат поиска сообщается на терминал.

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

Цель поиска ключевого слова в строке большего размера – определить первый индекс элемента в строке слов, начиная с которого все символы строки слов совпадают с символами ключевого слова. Для этого алгоритм поиска последовательно просматривает символы строки слов, проводя одновременное сравнение ее очередного символа с первым символом ключевого слова. После возникновения такого совпадения алгоритм производит последовательное сравнение соответствующих элементов ключевого слова и строки. При этом слово в строке должно быть отделено пробелами с двух сторон от других слов.

В данной программе мной были использованы главные операции над строками: **rep** (и его производные) – цикл повторений, **cmps** (и его производные) – сравнение строк и **scas** (и его производные) – сканирование строки. Все команды для работы со строками считают, что строка-источник находится по адресу **DS:SI**, то есть в сегменте памяти, указанном в **DS** со смещением в **SI**, а строка-приемник – в **ES:DI**. Также можно отметить, что все строковые инструкции работают только с одним элементом строки (байтом, словом (2 байта) или двойным словом (4 байта)).

В данной задаче мной было выделено 6 байт памяти под ключевое слово и 20 байт под строку.

3. Алгоритм решения поставленной задачи.

Данный алгоритм сравнивает слово и строку.

НАЧАЛО

взять первый символ строки

ПОВТОРИТЬ

ЕСЛИ текущий символ строки совпадает с первым символом слова ТО

ЕСЛИ все слово входит в строку ТО

ЕСЛИ с двух сторон пробелы (если только слово в строке не на 1-м и не на последнем месте) ТО
выделяем слово в строке и выдаем соответствующее сообщение

перейти к следующему символу в строке

ПОКА не закончится строка

КОНЕЦ

4. Тестирование программы.

Ключевое слово	Строка	Ответ программы
q	q	Да, слово нашлось
qw	qwe	Да, слово нашлось
qwe	aaa qwe aaa	Да, слово нашлось
qwe	qwe aaa	Да, слово нашлось
qwe	aaa qwe	Да, слово нашлось
q	l	Нет ключевого слова
qwe	qw	Слово длиннее строки
qwer	aaaqwey	Нет ключевого слова

- Если пользователь не ввел ключевое слово и нажал ENTER, то программа попросит ввести слово заново.
- Если пользователь вводит в слово пробел, то программа просит ввести слово заново.

```
;
;*****
;*****Лабораторная работа по ассемблеру №1*****
;*****на тему:"Обработка строк текста"*****
;*****Выполнил студент группы ВТ-21: Потеренко А.Г.*****
;*****Проверил преподаватель:Коноплянов А.В.*****
;*****Вариант №3*****
;*****
;Код размещается в одном сегменте, а данные и стек – в другом
.MODEL SMALL
;Описывает обычный сегмент данных
.DATA
;Массив, состоящий из слов (по 2 байта занимает каждый символ) – для окон
;координаты вертикальных линий у рамы
mas dw 326,486,646,806,966,1126,1286
;Переменная k-счетчик
k dw 2
;Массивы, состоящие из слов (по 2 байта занимает каждый символ) – для окон
;координаты вертикальных линий у рамы
mas1 dw 468,628,788,948,1108,1268,1428
mas2 dw 2086,2246,2406,2566,2726,2886,3046
mas3 dw 2228,2388,2548,2708,2868,3028,3188
mas4 dw 166,308,1446,1588,1926,2068,3206,3348
;Массив кодов символов "уголков для рамы"
mas5 dw 201,187,200,188,201,187,200,188
;Рабочие строки под псевдокомандой db:определяет байт для кода каждого символа
s1 db 'Laboratornaa Rabota Po Assembleru #1'
s2 db 'Student: Poterenko Aleksandr. BT-21'
s3 db 'Proveril Prepodavatel:KONOPLANOV A.B.'
s4 db 'BBEDITE KLUCHEBOE SLOBO <= 5 SIMBOLOV:'
s5 db 'BBEDITE SPISOK <=50 SIMBOLOV:'
s6 db 'SLOVO NAIDENNO'
s7 db 'NET KLUCHEBOGO SLOBA!!!'
s8 db 'PUSTO B KL. SLOVE'
s9 db 'PUSTO B STROKE'
s10 db 'SLOBO KOROCHE STROKI – NEBOSMOGNO SRABNIT!!!'
s11 db 'SLOBO NEMOGET SODERGAT PROBEL!!!'
k1 db 0
;Разрешены команды процессора 80186 – в нашей задаче для IMUL пр,ист1,ист2
.186
.386
;Директива описывает обычный сегмент стека, параметр 100h байт есть размер
;стека
.stack 100h
;*****
;*****Тело программы*****
;*****
.code
MAIN:
;В DS заносится сегментный адрес начала строки в .DATA
mov ax,@data
mov ds,ax
;Установка видеорежима, в AL номер режим (2-текстовый режим 80*25)
mov ah,00h
mov al,02h
int 10h
;В ES заносим адрес начала видеопамати
mov ax,0B800h
mov es,ax
;В DI заносим смещение относительно начала сегмента видеопамати, то есть
;на дисплее смещение относительно верхнего левого угла
mov di,166
;*****
;*****Рисование окон*****
;*****
;Рисование чистых синих окон без рамы и содержимого
;Значение 1297(счетчик окон – кол-во символов) определил подбором в отладчике
mov cx,1297
NEXT:
;Декремент регистра CX
```

```

dec cx
;Переход если CX=0
jcxz gorizortal
;В видеопамять записывается содержимое регистра AX:код и атрибут
;символа - "пусто" с атрибутом - синий на белом
mov ah,31
mov al,0ffh
mov es:[di],ax
;Увеличили DI на 2, так как каждый символ занимает 2 байта: код,атр
add di,2
;В AX записываем содержимое DS со смещением 0014 - там хранится наш
;счетчик
mov ax,ds:0014
;Формула на паскале ax:=ax*80+150
imul ax,ax,80
add ax,150
;Если DI=AX, то переходим на метку men
;Переход, если "Равно"
cmp di,ax
je men
JMP NEXT
;*****
NEN:
;Смещение переменной k равно 0014 относительно начала сегмента ds
mov ax,ds:0014
add ax,6
mov ds:[si+14],ax
;DI:=AX*80+6
imul ax,ax,80
add ax,6
mov di,ax
JMP NEXT
;*****
MEN:
;Если k=18, то переходим на метку nen
mov ax,ds:0014
cmp ax,18
je nen
;k:=k+2
mov ax,ds:0014
add ax,2
mov ds:[si+14],ax
;DI:=AX*80+6
imul ax,ax,80
add ax,6
mov di,ax
JMP NEXT
;*****
GORIZONTAL:;*****
;*****
;Самая верхняя горизонтальная линия
push 31
push 205
push 168
push 308
call smos
;Освобождаем 2 элемента стека, так как другие два на дне нам еще понадобятся
add sp,4
;*****
;Вторая линия горизонтальная
push 1448
push 1588
call smos
add sp,4
;*****
;Третья линия горизонтальная
push 1928
push 2068
call smos

```

```

add sp,4
;*****
;Четвертая линия горизонтальная
push 3208
push 3348
call smos
;Освобождаем стек от всех параметров GORIZONTAl включительно
add sp,8
JMP VERTICAL
;*****
;Главная процедура GORIZONTAl
SMOS PROC NEAR
push bp ;Дно стека увеличилось еще на 2 байта
mov bp,sp
mov ah,[bp+10]
mov al,[bp+8]
mov di,[bp+6]
CIKL:
    mov es:[di],ax
    add di,2
    cmp di,[bp+4]
    je METKA_PROC
JMP CIKL
METKA_PROC:
pop bp ;Освободили 2 байта, занимаемых переменной bp
ret ;Освободили 2 байта – адрес возврата для процедуры
SMOS ENDP
;*****
VERTICAL:;*****
;*****
;Проведение 4 вертикальных линий
mov ah,31
mov al,186
mov cx,7
mov si,offset mas
call smop
;*****
mov cx,7
mov si,offset mas1
call smop
;*****
mov cx,7
mov si,offset mas2
call smop
;*****
mov cx,7
mov si,offset mas3
call smop
JMP UGL
;*****
;Главная процедура VERTICAL
;Здесь SI-смещение массива MAS(i), а [SI]-то, что лежит по этому адресу
SMOP PROC NEAR
mov cx,7 ;Счетчик символов, в одной прямой их ровно 7
CIKL1:
    mov di,[si]
    mov es:[di],ax
    add si,2
    dec cx
    jcxz METKA_PROC1
JMP CIKL1
METKA_PROC1:
ret
SMOP ENDP
;*****
UGL:;*****
;*****
mov cx,8
mov si,0

```

```

CIKL8:
    mov di,mass4[si]
    mov ax,mass5[si]
    mov ah,31
    add si,2
    mov es:[di],ax
    dec cx
    jcxz BBOD
JMP CIKL8
;*****
;*****Вывод сообщений в окна программы*****
;*****
BBOD:
mov si,offset s1
push 364
push 36
call smod
add sp,4
;*****
mov si,offset s2
push 524
push 35
call smod
add sp,4
;*****
mov si,offset s3
push 684
push 37
call smod
add sp,4
;*****
mov si,offset s4
push 2088
push 38
call smod
add sp,4
;*****
mov si,offset s5
push 2408
push 29
call smod
add sp,4
JMP BBEDITE
;*****
SMOD PROC NEAR
mov ah,31
;Сохраняем регистр bp, так как он может использоваться в основной программе
push bp
mov bp,sp
mov di,[bp+6]
mov cx,[bp+4]
KOPIR:
;В AL кладем код первого символа строки s1 со смещением si
;При увеличении si в AL кладем последующие символы строки
mov al,[si]
;В видеопамять со смещением относительно начала B800h:DI записываем 2 байта
mov es:[di],ax
;Счетчики
inc si
add di,2
dec cx
;Если счетчик CX обнуляется переходим на метку
jcxz МЕТКА_PROC2
JMP KOPIR
МЕТКА_PROC2:
pop bp
ret
SMOD ENDP
;*****

```

```
;*****
;***Процедура бесконечного цикла ввода значений, обрывается если нажата ESC***
                        ANTIGLABN PROC
;*****
;*****Пользователь вводит ключевое слово*****
;*****
BBEDITE:
    mov ax,0B800h
    mov es,ax
    mov di,2248
    mov cx,5
    ;В данном случае мы используем буфер по адресу DS:0000, так как мы уже
    ;использовали этот буфер в начале программы, для экономии места мы
    ;используем его повторно (он нам уже не нужен, его затираем новыми зн.)
    mov si,0
BBED1:
    ;Функция ввода символа без отображения
    mov ah,08h
    int 21h
    ;Узнали, нажат ли пробел
    cmp al,20h
    je PROBELY
    ;Узнали, нажата ли TAB
    cmp al,9
    je BBED1
    ;Узнали, нажата ли SPACE
    cmp al,8
    je BBED1
    ;Узнали, нажата ли ESC
    cmp al,1bh
    je exit1
    ;Если AL=0, то есть некоторые управляющие клавиши
    cmp al,0h
    je RASHIR
    ;Узнали, нажата ли ENTER
    cmp al,0Dh
    je METKA_1
    ;Записываем ASCII код символа по адресу DS:SI
    mov ds:[si],al
    ;Отображение символов на дисплей - прямой ввод в видеопамять
    mov ah,31
    mov es:[di],ax
    ;Изменение счетчиков
    inc si
    add di,2
    dec cx
    ;Если CX=0, то переход на метку
    jcxz BBEP999
JMP BBED1
;*****
PROBELY:
    ;Адрес начала видеопамяти
    mov ax,0B800h
    mov es,ax
    ;Смещение в видеопамяти относительно начала строки
    mov di,2248
    mov si,offset s11
    mov cx,32
KOPIR30:
    mov al,[si]
    mov ah,31 ;Атрибут:символ на зеленом фоне
    mov es:[di],ax
    inc si
    add di,2
    dec cx
    jcxz METKA_30
JMP KOPIR30
METKA_30:
    mov ah,10h
```



```

int 16h
mov di,2248
mov cx,32
KOPIR31:
    mov al,20h
    mov ah,31 ;Атрибут:символ на зеленом фоне
    mov es:[di],ax
    inc si
    add di,2
    dec cx
    jcxz BBEDITE
JMP KOPIR31
;*****
METKA_1:
    cmp si,0h
    je METKA_2
JMP BBEDITE2
METKA_2:
CALL PUSTO
;*****
bbep999:
jmp bbep
;*****
;Расширенный ASCII код, то
RASHIR:
    ;Получить код символа, повторный вызов функции и в AL ASCII код
    int 21h
mov dh,10
;Цикл определяет нажата ли какая-нибудь из клавиш F1-F10
mov dl,59
CIKL_RASH:
    cmp al,dl
    je BBED1
    inc dl
    dec dh
    cmp dh,0
    je RASHIR2
JMP CIKL_RASH
RASHIR2:
    ;Если F11
    cmp al,133
    je BBED1
    ;Если F12
    cmp al,134
    je BBED1
    ;Если стрелка вверх
    cmp al,72
    je BBED1
    ;Если стрелка вниз
    cmp al,80
    je BBED1
    ;Если стрелка влево
    cmp al,75
    je BBED1
    ;Если стрелка вправо
    cmp al,77
    je BBED1
    ;Если INSERT
    cmp al,82
    je BBED1
    ;Если HOME
    cmp al,71
    je BBED1
    ;Если PAGE UP
    cmp al,73
    je BBED1
    ;Если PAGE DOWN
    cmp al,81
    je BBED1

```

```

;Если END
    cmp al,79
    je BBED1
;Если Delete
    cmp al,83
    je BBED1
;*****
;Если пользователь ввел в ключевом слове все 5 символов, то пока он не нажмет
;ENTER программа не перейдет ко вводу следующей строки
ВВЕР:
    ;Функция ввода символа без отображения в видеопамять
    mov ah,08h
    int 21h
    ;Узнали, нажата ли ESC
    cmp al,1bh
    je exit1
    ;Узнали, нажата ли ENTER
    cmp al,0Dh
    je BBEDITE2
JMP ВВЕР
;*****
;Выход №1 из программы при нажатии ESC
EXIT1:
    mov ah,00h
    mov al,02h
    int 10h
    mov ax,4c00h
    int 21h
;*****
;*****Пользователь вводит строку слов (не более 50 символов)*****
;*****
BBEDITE2:
    mov ax,0B800h
    mov es,ax
    mov di,2568
    mov cx,50
    push si
    ;Сохраняем адрес начала строки DS:SI
BBED2:
    ;Ввод символов, выход: AL=код символа
    mov ah,08h
    int 21h
    ;Узнали, нажата ли TAB
    cmp al,9
    je BBED2
    ;Узнали, нажата ли SPACE
    cmp al,8
    je BBED2
    ;Узнали, нажата ли ESC
    cmp al,1bh
    je exit2
    ;Если AL=0, то есть некоторые управляющие клавиши
    cmp al,0h
    je RASHIR3
    ;Узнали, нажата ли ENTER
    cmp al,0Dh
    je COPY
    ;Записываем ASCII код символа по адресу DS:SI
    mov ds:[si],al
    ;Отображение символов на дисплей - прямой ввод в видеопамять
    mov ah,31
    mov es:[di],ax
    ;Изменение счетчиков
    inc si
    add di,2
    dec cx
    ;Если CX=0, то переход на метку
    jcxz ВВЕР1
JMP BBED2
;*****

```

```

;Если пользователь ввел в строке все 50 символов, то пока он не нажмет
;ENTER программа не перейдет к вычислению, есть ли ключевое слово в строке
ВВЕР1:
    ;Функция ввода символа без отображения
    mov ah,08h
    int 21h
    ;Узнали, нажата ли ESC
    cmp al,1bh
    je exit2
    ;Узнали, нажата ли ENTER
    cmp al,0Dh
    je COPY
JMP ВВЕР1
;*****
;Расширенный ASCII код, то
RASHIR3:
    ;Получить код символа
    int 21h
mov dh,10
mov dl,59
CIKL_RASH1:
    cmp al,dl
    je BBED2
    inc dl
    dec dh
    cmp dh,0
    je RASHIR4
JMP CIKL_RASH1
RASHIR4:
    ;Если F11
    cmp al,133
    je BBED2
    ;Если F12
    cmp al,134
    je BBED2
    ;Если стрелка вверх
    cmp al,72
    je BBED2
    ;Если стрелка вниз
    cmp al,80
    je BBED2
    ;Если стрелка влево
    cmp al,75
    je BBED2
    ;Если стрелка вправо
    cmp al,77
    je BBED2
    ;Если INSERT
    cmp al,82
    je BBED2
    ;Если HOME
    cmp al,71
    je BBED2
    ;Если PAGE UP
    cmp al,73
    je BBED2
    ;Если PAGE DOWN
    cmp al,81
    je BBED2
    ;Если END
    cmp al,79
    je BBED2
    ;Если Delete
    cmp al,83
    je BBED2
;*****
;Выход №2 из программы при нажатии ESC
EXIT2:
    mov ah,00h

```

```

mov al,02h
int 10h
mov ax,4c00h
int 21h
;*****
;Вызов процедуры вывода сообщения о том, что строка пуста
МЕТКА_3:
CALL PUSTO2
;*****
;*****Подпрограмма поиска ключевого слова во введенной строке*****
;*****
COPY:
;Теперь SI=Смещение последнего символа в строке
dec si

mov ax,@data
mov es,ax

;BP=Смещение последнего символа в строке
mov bp,si

;Восстанавливаем SI = адрес начала строки DS:SI
pop si

;Так как у нас возврата из процедуры нет, то очищаем стек от точек входа
add sp,6

;DX=число символов в ключевом слове
mov dx,si

;Теперь SI=смещение последнего символа ключевого слова
dec si

;Теперь BP=число символов в строке
sub bp,si

;Если пользователь не ввел ни одного символа в строке
CMP bp,0h
JE МЕТКА_3

;Если ключевое слово короче строки (jl:если меньше)
cmp bp,dx
jl KOROCHE

GLAVNCICL:
;В AL заносится первый код символа ключевого слова для дальнейшего поиска
mov al,ds:0000

;Если вся строка прочитана и не найдено ни одного сходства, то выход
cmp si,bp
je ОТВЕТ_NET

;BP=Длина всей строки минус смещение последнего символа ключевого слова
;SI в данном случае будет увеличиваться
PUSH bp
SUB bp,si
;Счетчик цикла: пробегание циклом всей строки минус число символов ключевого
;и плюс 1
mov cx,bp
POP bp

;Увеличение регистра на 1
inc si

;В DI заносится смещение первого символа строки
mov di,si

;Сканирование строки: cx уменьшается каждый раз, когда выполняется цикл
;сравнивает содержимое регистра со строкой

```

```

repne scasb

;После выполнения операции DI увеличился, поэтому мы его уменьшаем и теперь
;DI=смещение первого совпадающего символа
dec di

;Теперь DX равно порядковому номеру в строке совпадающего символа
push dx
push di;Сохраняем временно DI, чтобы не менять его значение в осн. программе
inc di
SUB di,dx
;Записали в переменную k1 номер совп. символа в строке, чтобы выделить его
PUSH SI
mov si,offset k1
mov ds:[si],di
POP SI
pop di ;Восстанавливаем DI, как-будто мы его и не использовали
pop dx ;Восстанавливаем DX, опять DX=число символов в ключевом слове

;Дальнейший поиск:продолжает сравнивать остальную часть ключевого слова
;с тем, которое введено в строке
;В CX заносится число символов в ключевом слове (счетчик 2-го цикла ср-ния)
mov cx,dx
inc cx

;Сохранение DI и SI
push di
push si
;В SI заносится смещение второго символа, так как мы сравниваем DS:SI
;с ES:DI
mov si,0
;Операция сравнения остальных символов DS:SI и ES:DI
repe cmpsb
;Восстановление SI до прежнего значения
pop si
pop di

;Если CX=0, значит данное ключевое слово присутствует в строке
jcxz METKA_19
JMP GLAVNCICL
;*****
;Если слова совпали, то необходимо проверить есть ли пробелы сзади и спереди
METKA_19:
;Если найденное слово стоит первым в строке, то проверяем пробел справа
cmp di,dx
je METKA_17
;Если найденное слово стоит последним в строке, то проверяем пробел слева
PUSH DX
PUSH BP
PUSH DI
dec dx ;Теперь DX= смещение последнего символа ключевого слова
add bp,dx ;Теперь BP=смещение последнего символа строки
add di,dx ;Теперь DI=смещение 1-го совпадающего символа+dx
cmp di,bp
POP DI
POP BP
POP DX
je METKA_18
;Если найденное слово стоит посередине, то проверяем оба пробела
JMP METKA_20
;*****
METKA_17: ;Еще нужно проверить строка состоит из одного слова или нет. Если
;нет, то на метку METKA_21
cmp dx,bp ;Равны ли строка и ключевое слово по количеству символов
je ОТВЕТ_ДА
JMP METKA_21 ;Если нет то проверяем пробел справа
;*****
METKA_18:
;Смысла нет сохранять регистры, т.к. это сравнение последнее

```

```

    dec di ;Теперь DI=смещение 1-го совпадающего символа в строке минус 1
    mov si,di
    mov ax,[si]
    cmp al,20h
    je OTBET_DA
    JMP OTBET_NET ;Все равно слово стоит последним -> смысла нет дальше провер.
;*****
METKA_20: ;Проверяем пробел слева, если найд. слово в строке не на 1 месте
    PUSH SI
    PUSH DI
    dec di
    mov si,di
    POP DI
    mov ax,[si]
    POP SI
    cmp al,20h
    JE METKA_21
    JMP GLAVNCICL
;*****
METKA_21: ;Проверяем пробел справа
    PUSH SI
    PUSH DI
    add di,dx
    mov si,di
    POP DI
    mov ax,[si]
    POP SI
    cmp al,20h
    je OTBET_DA
;Если все выше не подходит, то продолжаем поиск
JMP GLAVNCICL
;*****
;*****Сообщения для вывода*****
;*****
;*****Если строка короче ключевого слова*****
;*****
KOROCHE:
;Адрес начала видеопамати
mov ax,0B800h
mov es,ax
;Смещение строки s10 относительно начала сегмента
mov si,offset s10
mov di,2728
mov cx,44
KOPIR10:
    mov al,[si]
    mov ah,31
    mov es:[di],ax
    inc si
    add di,2
    dec cx
    jcxz METKA_10
JMP KOPIR10
METKA_10:
CALL STIR
;*****
;*****Если не найденно ключевого слова*****
;*****
OTBET_NET:
;Адрес начала видеопамати
mov ax,0B800h
mov es,ax
;Смещение строки s7 относительно начала сегмента
mov si,offset s7
mov di,2728
mov cx,23
KOPIR7:
    mov al,[si]
    mov ah,31

```

```

        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_9
JMP KOPIR7
METKA_9:
CALL STIR
;*****
;*****Если ответ "Да", то выводим соответствующее сообщение*****
;*****
ОТВЕТ_ДА:
;Адрес начала видеопамати
mov ax,0B800h
mov es,ax
;Смещение строки s6 относительно начала сегмента
mov si,offset s6
mov di,2728
mov cx,14
KOPIR6:
        mov al,[si]
        mov ah,31
        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_4
JMP KOPIR6
METKA_4:
;*****
;Ответ "Да", значит вызываем процедуру выделения ключевого слова в строке
CALL BIDILENIE
;*****
;*****Главный выход из программы*****
;*****
EXIT:
;Программа ожидает нажатия какой-либо клавиши: функция №10 прерывания 16h
mov ah,10h
int 16h
;Переход в ведеорежим 80/25: мы таким образом очищаем видеопамать (дисплей)
mov ah,00h
mov al,02h
int 10h
;Функция Доса корректно завершает программу
mov ax,4c00h
int 21h
;*****
;*****Если пусто в ключевом слове*****
;*****
PUSTO PROC
;Адрес начала видеопамати
mov ax,0B800h
mov es,ax
;Смещение строки s8 относительно начала сегмента
mov si,offset s8
mov di,2248
mov cx,17
KOPIR8:
        mov al,[si]
        mov ah,31
        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_8
JMP KOPIR8
METKA_8:
CALL STIR
PUSTO ENDP

```

```
;*****
;*****Если пусто в строке*****
;*****
PUSTO2 PROC
;Адрес начала видеопамяти
mov ax,0B800h
mov es,ax
;Смещение строки s9 относительно начала сегмента
mov si,offset s9
mov di,2568
mov cx,14
KOPIR9:
    mov al,[si]
    mov ah,31
    mov es:[di],ax
    inc si
    add di,2
    dec cx
    jcxz METKA_6
JMP KOPIR9
METKA_6:
CALL STIR
PUSTO2 ENDP
;*****
;*****Процедура выделения найденного ключевого слова в строке*****
;*****
BIDILENIE PROC
;Адрес начала видеопамяти
mov ax,0B800h
mov es,ax
;Смещение в видеопамяти относительно начала строки
mov di,2568
;*****
PUSH DX ;Сохраняем DX в стеке
mov si,offset k1
mov dx,[si] ;В регистр dx заносим значение k1:на каком месте стоит совпадающий
;символ в строке
dec dx
imul dx,2 ;Умножение на 2, так как в видеопамяти каждый символ по 2 байта
add di,dx ;Заносим в DI значение смещения в видеопамяти
POP DX ;Восстанавливаем DX
;*****
mov cx,dx ;Счетчик, сколько букв в ключевом слове
mov si,0 ;Так как мы просто копируем ключевое слово с адреса ds:0000
KOPIR11:
    mov al,[si]
    mov ah,32 ;Атрибут:символ на зеленом фоне
    mov es:[di],ax
    inc si
    add di,2
    dec cx
    jcxz METKA_7
JMP KOPIR11
METKA_7:
CALL STIR
BIDILENIE ENDP
;*****
;*****Процедура затирания полей для ввода, для повторного ввода*****
;*****
STIR PROC
;Программа ожидает нажатия какой-либо клавиши: функция №10 прерывания 16h
mov ah,10h
int 16h
;Адрес начала видеопамяти
mov ax,0B800h
mov es,ax
mov di,2248
mov cx,51
KOPIR12:
```



```

        mov al,0FFh ;Код символа есть "пробел"
        mov ah,31
        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_4_1
JMP KOPIR12
METKA_4_1:
mov di,2568
mov cx,51
KOPIR13:
        mov al,0FFh ;Код символа есть "пробел"
        mov ah,31
        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_5
JMP KOPIR13
METKA_5:
mov di,2728
mov cx,51
KOPIR14:
        mov al,0FFh ;Код символа есть "пробел"
        mov ah,31
        mov es:[di],ax
        inc si
        add di,2
        dec cx
        jcxz METKA_6_1
JMP KOPIR14
METKA_6_1:
CALL ANTIGLABN ;ВЫЗОВ ПРОЦЕДУРЫ ДЛЯ ПОВТОРНОГО ВВОДА
STIR ENDP
;*****
;*****КОНЕЦ ПРОЦЕДУРЫ ОСНОВНЫХ ДЕЙСТВИЙ ПРОГРАММЫ*****
        ANTIGLABN ENDP
;*****
END MAIN

```

```

;*****
;*****ЛР №1 Версия №2*****
;*****
.MODEL SMALL
.DATA
lenslobo equ 6
lenstroka equ 20
so db 6 dup (' ')
sol db 20 dup (' ')
k1 db ? ;Считает сколько символов в ключевом слове
k2 db ? ;Считает сколько символов в строке
k3 dw ? ;Считает с какой позиции произойдет сканирование строки (es:di=k3)
s1 db 'Введите слово:',0DH,0Ah,'$'
s2 db 'Введите строку:',0DH,0Ah,'$'
s3 db 'Увы, ключевого слова не нашлось$'
s4 db 'Да, слово есть в строке$'
s5 db 'Слово длиннее строки$'
perenos db 0DH,0Ah,'$'
.STACK 100h
.CODE
;*****
MAIN:
    mov ah,00h
    mov al,02h
    int 10h
    mov ax,@data
    mov ds,ax
    lea dx,s1
    mov ah,9
    int 21h
;*****
    xor si,si
    mov cx,lenslobo
    mov dh,1 ;Для курсора
    mov dl,0 ;Для курсора
SLOBO_LOOP:
    mov ah,08h
    int 21h
;*****Расширенная клавиатура*****
    cmp al,0
    jne PROBEL
    int 21h
    JMP SLOBO_LOOP
;*****Пробел*****
PROBEL:
    cmp al,20h
    je SLOBO_LOOP
;*****
    cmp al,0dh
    je GOTO_BBOD_STROKA
;*****
    cmp al,27
    jne SLED
    jmp near:exit
;*****Буквы английского алфавита*****
SLED:
    cmp al,65
    jge ok
    jmp SLOBO_LOOP
ok: cmp al,122
    jle ok_2
    jmp SLOBO_LOOP
ok_2:;Район от 91 до 96 - ненужные знаки
    cmp al,91
    jge ok_3
    jmp ok_4
ok_3:cmp al,96
    jle SLOBO_LOOP
;*****

```

```

        ok_4:
        push cx
        mov cx,1
        mov bl,7
        mov ah,9h
        int 10h
        pop cx
        ;*****
        mov ds:[si],al
        inc si
        inc k1
        ;*****Курсор*****
        mov ah,02h
        inc dl
        int 10h

        LOOP SLOBO_LOOP
;*****
GOTO_BBOD_STROKA:
        ;Приглашение ко вводу строки
        lea dx,perenos
        mov ah,9
        int 21h
        lea dx,s2
        mov ah,9
        int 21h
        ;*****
        mov dh,3          ;Для курсора
        mov dl,0          ;Для курсора
        ;*****
        mov cx,lenstroka
        lea si,sol
        CIKL_BBODA_STROKA:
        mov ah,08h
        int 21h
        ;*****Расширенная клавиатура*****
        cmp al,0
        jne PROBEL_1
        int 21h
        JMP CIKL_BBODA_STROKA
        ;*****
        PROBEL_1:
        cmp al,0dh
        je GO_SRABNENIE
        ;*****
        cmp al,27
        jne SLED_1
        jmp near:exit
        ;*****Буквы английского алфавита*****
        SLED_1:
        cmp al,65
        jge ok1
        jmp CIKL_BBODA_STROKA
        ok1: cmp al,122
        jle ok1_2
        jmp CIKL_BBODA_STROKA
        ok1_2:;Район от 91 до 96 - ненужные знаки
        cmp al,91
        jge ok1_3
        jmp ok1_4
        ok1_3:cmp al,96
        jle CIKL_BBODA_STROKA
        ;*****
        ok1_4:
        push cx
        mov cx,1
        mov bl,7
        mov ah,9h
        int 10h
        pop cx

```

```

;*****
mov ds:[si],al
inc si
inc k2
;*****Курсор*****
mov ah,02h
inc dl
int 10h
LOOP CIKL_BBODA_STROKA
;*****
GO_SRAVNENIE:
;*****Если вообще ключевое слово длиннее строки*****
PUSH cx
PUSH dx
MOV cl,k1
MOV dl,k2
CMP cl,dl
POP dx
POP cx
JG KOROCHE
JMP GO_ONE
KOROCHE:
lea dx,perenos
mov ah,9
int 21h
lea dx,s5
mov ah,9
int 21h
MOV ah,10h
INT 16h
JMP NEAR:EXIT
;*****Если и строка и слово по одному символу*****
GO_ONE:
MOV cl,k1
MOV dl,k2
CMP cl,1
JNE GO_POISK
CMP dl,1
JNE GO_POISK
LEA si,so
MOV al,[si]
LEA si,so1
MOV ah,[si]
CMP al,ah
JE OTBET_DA
JMP OTBET_NET
;*****
GO_POISK:
MOV ax,@data
MOV es,ax
;*****
LEA si,so
MOV al,[si]
;*****
XOR ch,ch ;Обнуляем старший байт в регистре, работаем с cl
MOV k3,offset so1
GLABN_CIKL:
MOV cl,k2 ;В регистр заносится количество символов в строке
MOV di,k3
REPNE SCASB ;Теперь cx=k2-p (p=порядковый номер в строке совп. симв.)
MOV k3,di ;Сохраняем, с какой позиции в след. раз начнем сканиров.
MOV k2,cl ;Сохраняем количество оставшихся в строке неотск. симв.
DEC di ;Теперь DI=смещение первого совпадающего символа в строке
;*****
CMP cx,0
JE OTBET_NET
;*****
MOV cl,k1
INC cl

```

```

        LEA si,so                ;В si кладем начало адреса ключевого слова
        REPE CMPSB              ;Строка:ES:DI, а слово по адресу DS:SI
        JE OTBET_DA
        ;*****Счетчики*****
        INC SI
JMP GLABN_CIKL
;*****
OTBET_NET:
        LEA dx,perenos
        MOV ah,9
        INT 21h
        LEA dx,s3
        MOV ah,9
        INT 21h
        JMP EXIT
;*****
OTBET_DA:
        LEA dx,perenos
        MOV ah,9
        INT 21h
        LEA dx,s4
        mov ah,9
        INT 21h
;*****
        MOV ah,10h
        INT 16h
EXIT:
        MOV ah,4ch
        INT 21h
END MAIN

```