TITLE FAIL(EXE)

assume CS:code, DS:data

; описываем сегмент кода

code segment

PROGRAM\_START: ; метка точки входа программы

; настройка регистра DS

mov AX, data

mov DS, AX

; освобождение памяти

mov AX, abcd ; сегментный адрес конца программы

mov DX, ES ; сегментный адрес начала программы

sub AX, DX ; размер программы в параграфах

mov BX, AX ; оставим его в BX

mov AH, 4ah ; функция изменения размера блока

int 21h

; инициализируем графический режим : начало

mov AH, 00 ; функция установки граф. режима

mov AL, 0dh ; устанавливаем разрешение 320х200

int 10h ; вызов BIOS

CALL Q10CLR ;вызов подпрограммы очистки экрана

CALL T10SCR ;вызов подпрограммы перекрашивания экрана в другой цвет

MOV ROW,10 ;выбор строки

MOV COL,13 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

LEA SI,COUT1 ;передаем в регистр SI адрес начала строки

MOV COUNT,16 ;устанавливаем длину строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,11 ;выбор строки

MOV COL,4 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

LEA SI,COUT2 ;передаем в регистр SI адрес начала строки

MOV COUNT,32 ;устанавливаем длину строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,12 ;выбор строки

MOV COL,8 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,24 ;устанавливаем длину строки

LEA SI,COUT3 ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,13 ;выбор строки

MOV COL,8 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,24 ;устанавливаем длину строки

LEA SI,COUT4 ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,14 ;выбор строки

MOV COL,15 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,10 ;устанавливаем длину строки

LEA SI,COUT5 ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV AH,7h ;задержка ожидания

INT 21h ;ввода символа

CALL Q10CLR ;вызов подпрограммы очистки экрана

CALL T10SCR ;вызов подпрограммы перекрашивания экрана в другой цвет

MOV ROW,10 ;выбор строки

MOV COL,01 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

LEA DX,PATH1 ;DS:DX указывают на старый путь

MOV AX,SEG PATH2 ;ES:DI указывают на новый путь

MOV ES,AX ;заносим его в ES

LEA DI,PATH2 ;указываем новое имя файла и путь

MOV AH,56H ;номер функции

INT 21H ;переносим файл

JC ERROR ;уход на обработку ошибки

MOV AH,43h ;номер функции изменения атрибута файла

MOV AL,01h ;изменяем атрибут файла

MOV CX,0021h ;код атрибута только чтение

LEA DX,PATH2 ;в DX заносим путь к файлу

INT 21h ;вызов DOS

MOV COUNT,32 ;устанавливаем длину строки

LEA SI,MESSAGE ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,11 ;выбор строки

MOV COL,01 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,16 ;устанавливаем длину строки

LEA SI,MESSAGE1 ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

MOV ROW,13 ;выбор строки

MOV COL,01 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,32 ;устанавливаем длину строки

LEA SI,MESSAGE2 ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

JMP A

ERROR:

MOV ROW,15 ;выбор строки

MOV COL,01 ;выбор столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

MOV COUNT,38 ;устанавливаем длину строки

LEA SI,ERRORSTRING ;передаем в регистр SI адрес начала строки

CALL B10PRMP ;вызов подпрограммы вывода строки на экран

A:

mov ah,7

int 21h

INT 20h ;вызов прерывание окончания работы программы

B10PRMP PROC NEAR ;подпрограмма вывода строки на экран

B20:

MOV BL,171 ;установка атрибутов текста

CALL F10DISP ;вызов подпрограммы вывода сивола на экран

INC SI ;увеличение на единицу адреса смещения начала строки

INC COL ;увеличение на единицу столбца

CALL Q20CURS ;вызов подпрограммы установки курсора

DEC COUNT ;уменьшения на единицу количества выводимых символов

JNZ B20 ;переход к метке В20 если еще есть символы

RET ;возврат

B10PRMP ENDP ;конец подпрограммы

F10DISP PROC NEAR ;подпрограмма вывода символа на экран

MOV AH,09 ;функция вывода на экран

MOV AL,[SI] ;заносим в AL код выводимого символа

MOV BH,00 ;выбор страницы

MOV CX,01 ;число повторений

INT 10h ;вызов BIOS

RET ;возврат

F10DISP ENDP ;конец подпрограммы

Q20CURS PROC NEAR ;подпрограмма установки курсора

MOV AH,02 ;функция установки курсора

MOV BH,00 ;выбор страницы

MOV DH,ROW ;выбор строки

MOV DL,COL ;выбор столбца

INT 10h ;вызов BIOS

RET ;возврат

Q20CURS ENDP ;конец подпрограммы

Q10CLR PROC NEAR ;подпрограмма очистки экрана

MOV AX,0600h ;AH=06 (прокрутка), AL=00 (весь экран)

MOV BH,07h ;нормальный атрибут

MOV CX,0000h ;верхняя левая позиция

MOV DX,184Fh ;нижняя правая позиция

INT 10h ;вызов BIOS

RET ;возврат

Q10CLR ENDP ;конец подпрограммы

T10SCR PROC NEAR ;подпрограмма перекрашивания экранав другой цвет

MOV AH,06h ;прокрутить экран вверх

MOV AL,01h ;на одну строку

MOV BH,25h ;выбор атрибута

MOV CH,00h ;номер строки левого верхнего угла

MOV CL,00h ;номер столбца левого верхнего угла

MOV DH,18h ;номер строки нижнего правого угла

MOV DL,4Fh ;номер столбца нижнего правого угла

INT 10h ;вызов BIOS

RET ;возврат

T10SCR ENDP ;конец подпрограммы

code ends ;конец сегмента кода

; описание сегмента данных (переменные)

data segment

COUNT DB ?

COUT1 DB " Rabota"

COUT2 DB "Sistemnoe programnoe obespecenie"

COUT3 DB "Studenta gruppi "

COUT4 DB "ФИО"

COUT5 DB "Variant #3"

MESSAGE DB "Fail C:\NEWFILE.TXT pereimenovan"

MESSAGE1 DB "v C:\OLDFILE.TXT"

MESSAGE2 DB "Ustanovlen atribut Tolko chtenie"

ERRORSTRING DB "Oshibka: fail C:\NEWFILE.TXT ne naiden"

PATH1 DB 'C:\NEWFILE.txt',0

PATH2 DB 'C:\OLDFILE.txt',0

COUT DB ?

ROW DB 00

COL DB 00

data ends

; описываем сегмент стека

stk segment stack

db 256 dup (?) ; выделяем под стек 256 байт

stk ends

abcd segment 'endseg' ; фиктивный сегмент для определения размера программы

abcd ends

end PROGRAM\_START ; конец программы