Крючков Илья, ИВТ-31

Задание:

Вариант 32.

Ввести строку символов. Вывести строку в разных частях экрана разным цветом (текстовый режим)

Исходный код:

```
; Крючков Илья, ИВТ-31
; Вариант 32
; Ввести строку символов. Вывести строку в разных частях экрана
; разным цветом (текстовый режим)
;

.model small
.data
     message0 db 'String: $'
     string0 db 10 DUP(?)
     color db 1 ; цвет строки
     ltpx db 1 ; позиция строки по X
     ltpy db 1 ; позиция строки по Y
     vmode db ? ; для сохранения видеорежима
     charcnt dw 0 ; кол-во введенных символов
     .stack 256h ; размер стека

.code

printMessage proc near ; процедура вывода сообщений
     push ax ; сохраняем значение регистров ax
     push cx ; и cx

     mov ah, 09h ; функция вывода строки на экран
     int 21h ; dos прерывание
```

```
	pop cx ; восстановление значений регистров cx
	pop ax ; и ax
	ret ; возврат из подпрограммы
printMessage endp


getStr proc near ; процедура: ввод строки в [bx]
	push cx ; занесение cx в стек
	push dx ; занесение dx в стек

	mov cx, 0Ah ; кол-во символов в строке
	mov dx, 01h ; enter не учитывается

getstr_loop:
	call getChar ; получить след. символ
	cmp byte ptr [bx], 0Dh ; проверка на enter
	je getstr_quit ; если enter, переход на метку getstr_quit
	inc bx ; иначе, увеличение адреса, для сохранения след. символа
	inc charcnt ; увеличение кол-ва символов
	loop getstr_loop ; переход к след. итерации, либо завершение
цикла

getstr_quit:
	pop dx ; возврат из стека dx
	pop cx ; возврат из стека cx
	ret ; возврат из процедуры

getStr endp

getChar proc near ; ввод символа в [bx]
	push ax ; занесение ax в стек
```

```
	push dx ; занесение dx в стек

getchar_m1:
	mov ah, 08h ; считывание символа без эха
	int 21h ; функцией dos

	cmp al, 00h ; проверка на функциональную клавишу
	jne getchar_m2 ; если не fn, переход к метке getchar_m2

	mov ah, 08h
	int 21h
	jmp getchar_m1

getchar_m2:
	cmp dx, 01h ; учитывается ли enter
	jne getchar_m3 ; если не учитывается, переход на getchar_m3

	cmp al, 0Dh ; если enter
	je getchar_write ; переход в конец

getchar_m3:
	cmp al, 31 ; проверка на управляющий символ
	jle getchar_m1 ; если код клавиши меньше или равен 31, переход на
getchar_m1

getchar_write:
	mov [bx], al ; сохраняем в буфер символ
	mov dl, al ; символ в dl
	mov ah, 02h ; функция dos вывода на экран
	int 21h ; вывод символа

	pop dx ; возврат из стека dx
```

```
	pop ax ; возврат из стека ax
	ret ; возврат из процедуры

getChar endp

drawText proc near ; процедура вывода текста
	push ax ; занесение ax в стек
	push bx ; занесение bx в стек
	push cx ; занесение cx в стек

	; получить X
	; параметры для процедуры
	mov ah, 08h
	mov al, ltpx
	mov bl, 81
	call rand ; вызов процедуры получения пвевдослучайного числа [0,
81)
	mov ltpx, ah ; полученное значение X

	; получить Y
	; параметры для процедуры
	mov ah, 07h
	mov al, ltpy
	mov bl, 26
	call rand ; вызов процедуры получения пвевдослучайного числа [0,
26)
	mov ltpy, ah ; полученное значение Y


	mov ah, 13h ; номер функции
	mov al, 01h ; оставить курсор в начале
	mov cx, charcnt ; длина строки
```

```
	mov bl, color ; цвет
	mov dl, ltpx ; X
	mov dh, ltpy ; Y
	lea bp, string0 ; загрузка эффективного адреса первого байта строки в bp
	int 10h ; прерывание BIOS

	; установить следующий цвет
	; параметры для процедуры
	mov ah, 03h
	mov al, color
	mov bl, 10h
	call rand ; вызов процедуры получения пвевдослучайного числа [0, 16)

	mov color, ah ; полученное значение

	pop cx ; возврат из стека cx
	pop bx ; возврат из стека bx
	pop ax ; возврат из стека ax
	ret ; возврат из процедуры

drawText endp


clearSrc proc near ; процедура очистки экрана (переустановка видеорежима)
	mov ah, 00h
	mov al, 03h ; текстовый режим
	int 10h
	ret
clearSrc endp
```

```
; получить случайное число (0, X)
; Xn+1 = (aXn-1 + c) mod m
; Xn-1 - в al
; a - в ah
; m (max X) - в bl
; result: ah
rand proc near
	mul ah ; умножение значения из al на значение из ah, рез-т в ax
	add ax, 03h ; суммирование значения из ax и 03h, рез-т в ax
	div bl ; деление значения из ax на значение из bl, рез-т в al, остаток в ah
	ret
rand endp

main:
	mov ax, @data ; установка в ds и es
	mov ds, ax ; адреса сегмена данных
	mov es, ax

	lea dx, message0 ; загрузка эфф. адреса сообщения
	call printMessage ; вывод сообщения

	lea bx, string0
	call getStr	; ввод строки

	; считывание видеорежима
	mov ah, 0fh
	int 10h
	mov vmode, al

	; основной цикл прграммы
```

```
	main_loop:
		call clearSrc ; вызов процедуры очистки экрана
		call drawText ; вызов процедуры вывода текста

		; считывание клавиши
		mov ah, 00h
		int 16h

		cmp al, 0Dh ; если нажали enter
		je main_loop ; продолжаем цикл, иначе выход из цикла

	; сброс видеорежима
	mov ah, 00h
	mov al, vmode
	int 10h

	mov ah, 4Ch ; функция закрытия программы
	int 21h

end main
```