ORG 0x0

V0: WORD $default, 0x180 ; Вектор прерывания #0

V1: WORD $default, 0x180 ; Вектор прерывания #1

V2: WORD $int2, 0x180 ; Вектор прерывания #2

V3: WORD $int3, 0x180 ; Вектор прерывания #3

V4: WORD $default, 0x180 ; Вектор прерывания #4

V5: WORD $default, 0x180 ; Вектор прерывания #5

V6: WORD $default, 0x180 ; Вектор прерывания #6

V7: WORD $default, 0x180 ; Вектор прерывания #7

default: IRET ; обработка прерывания по умолчанию

ORG 0x029

X: WORD 0x0016 ; X = +26 (начальное значение)

max: WORD 0x001A ; max = +26

min: WORD 0xFFE8 ; min = -24

tmp: WORD 0x0 ; временная ячейка

org 0x40

start: DI ; запрет прерываний

cla ; очистка аккумулятора

out 0x1 ; запрет прерываний

out 0x3

out 0xA

out 0xD

out 0x11

out 0x15

out 0x19

out 0x1D ; последний запрет прерываний

ld #0xA ; (1000|0010) = 1010 = 0xA

out 0x5

ld #0xB ; (1000|0011) = 1011 = 0xB

out 0x7

EI

main: DI ; запрет прерываний

ld $X ; загрузка значения X

dec ; X = X - 1

call check

st $X ; сохранение нового значения X

HLT

EI ; разрешение прерываний

jump main ; переход в начало цикла

int2: ; Обработка прерывания от ВУ-2

in 0x4 ; Чтение из РД ВУ-2

st $tmp ; Сохранение Y во временный регистр

add $tmp ; Y + Y = 2Y

add $tmp ; 2Y + Y = 3Y

neg ; -3Y

add $X ; X = X - 3Y

call check ; Проверка выхода за ОДЗ

st $X ; Сохранение нового значения X

HLT

iret ; Возврат из обработки прерывания

int3:

; DI ; запрет прерываний

; push ; сохранение AC в стек

ld $X ; загрузка X

neg ; -X

sub $X ; -2X

sub $X ; -3X

sub $X ; -4X

sub $X ; -5X

add #6 ; F(X) = -5X + 6

out 0x6 ; вывод на ВУ-3

HLT

; pop ; восстановление AC из стека

; EI ; разрешение прерываний

IRET ; возврат из прерывания

; корректно

check:

check\_min:

cmp min ; сравнение с min

bpl check\_max ; если >= min → проверить max

ld $max ; если < min → установить max

jump fini

check\_max:

cmp max ; сравнение с max

bmi fini ; если <= max → всё в порядке

ld\_max:

ld $max ; если > max → установить max

; корректно

fini:

ret ; возврат