.686

.model flat, stdcall

.stack 100h

.data

X dw 5429h

Y dw 7844h

Z dw 0AD43h

Q dw 5622h

R dw 0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

; Шаг 1: Инкремент и суммирование

mov esi, offset X ; Загрузка адреса X

mov ecx, 4 ; Счетчик цикла

xor ebx, ebx ; Обнуление суммы L

loop\_inc:

mov ax, [esi] ; Загрузка значения

inc ax ; Инкремент

mov [esi], ax ; Сохранение обратно

add bx, ax ; Добавление к сумме L

add esi, 2 ; Следующий элемент

loop loop\_inc

; Шаг 2: Вычисление M

mov ax, bx ; AX = L

and ax, [X] ; L & X'

mov dx, ax

mov ax, bx

and ax, [Y] ; L & Y'

sub dx, ax ; DX = M

; Выбор подпрограммы

cmp dx, 921Bh

jae call\_sp1 ; Если M >= 921Bh

call subprog2 ; Если M < 921Bh

jmp check\_even

call\_sp1:

call subprog1

check\_even:

; Шаг 3: Проверка четности R

mov ax, [R]

test ax, 1

jz even\_r

jmp odd\_r

even\_r:

or ax, 009Fh ; R OR 009Fh

mov [R], ax

jmp ADDR1

odd\_r:

dec ax ; R - 1

mov [R], ax

jmp ADDR2

subprog1 PROC ; Подпрограмма 1

mov ax, dx

shr ax, 1 ; M / 2

sub ax, 12B9h

mov [R], ax

ret

subprog1 ENDP

subprog2 PROC ; Подпрограмма 2

mov ax, [Q]

shr ax, 1 ; Q'/2

mov bx, dx

sub bx, ax ; M - Q'/2

mov [R], bx

ret

subprog2 ENDP

ADDR1: ; Код для четного R

; ... (Дополнительные операции)

jmp exit

ADDR2: ; Код для нечетного R

; ... (Дополнительные операции)

exit:

Invoke ExitProcess, 0

End Start