Даны натуральные числа n, q1…qn, найти те члены q1 последовательности q1…qn, которые являются удвоенными нечетными числами.

Accumulator register (AX) - для арифметических операций.  
Counter register (CX) - для сдвигов и циклов.  
Data register (DX) - для арифметических операций и операций ввода/вывода.

Base register (BX) - для указателя на данные.  
Stack Pointer register (SP) - для указателя вершины стека.  
Stack Base Pointer register (BP) - для индикатора основания стека.  
Source Index register (SI) - для указателя отправителя (источника).  
Destination Index register (DI) - для получателя.

<https://vk.com/@physics_math-yazyk-assembler-komandy-i-osnovy-assemblera>

format MZ

org 100h

; main part

// В качестве параметра *call print\_str* передаётся адрес строки в регистре DI. Строка должна оканчиваться символом ‘$’, на экран выводится строка s\_in\_n

mov di, s\_in\_n ; В *di* вносится текст для вывода

call print\_str ; Выводит на экран

call input ; Вносит в ax число до 255

mov [n], ax ; Вносим переменную n в dx

call print\_endline

mov si,0

mov ebx,0

mov cx,2

MainLoop:

mov di, s\_in\_a ; В di вносится текст для вывода

call print\_str ; Выводит на экран

call input ; Вызов процедуры ввода числа

call print\_endline ; Вывод конца строки

push ax ; заносим в стек значение регистра ax

mov dx,0 ; в dx заносим 0

div cx ; деление ax/cx

cmp dx,0 ; сравниваем dx и 0

je L1 ; если они оба равны то переход в L1

jmp L3 ;прыжок в L3

L1:

mov dx,0

div cx

pop ax

cmp dx,1

je L2

jmp L3

L2:

mov [res+2\*ebx], ax

inc ebx

L3:

inc si

cmp si, [n]

jb MainLoop

mov di, s\_out ; В di вносится текст для вывода

call print\_str ; Выводит на экран

call print\_endline

mov esi,0

OutLoop:

mov ax, [res+2\*esi]

call print\_word\_sdec

call print\_endline

inc esi

cmp esi, ebx

jb OutLoop

mov di, s\_pak

call print\_str ; 'Press any key...'

mov ah,8 ; Функция DOS 08h - Ввод символа без эха

int 21h

mov ax,4C00h

int 21h ; Завершение программы

input:

push bx

push cx

mov ah,01h

int 21h ; В al первый символ

sub al,30h ; Теперь первая цифра

mov ah,0 ; Расширение до word

mov bx,10

mov cx, ax ; В cx первая цифра

Loop1:

mov ah,01h

int 21h ; В al Следующий символ

cmp al,0dh ; Сравнение с символом Enter

je Exit ; Конец ввода

sub al,30h ; В al - Следующая цифра

cbw ; Расширение до word

xchg ax,cx ; Теперь в ax - предыдущее число, в cx следующее

mul bx ; ax\*10

add cx,ax ; cx=ax\*10+cx

jmp Loop1 ; Продолжение ввода

Exit:

mov ax,cx

pop cx

pop bx

ret ; Возвращаемся к коду вызова функции (call) из основного кода

word\_to\_udec\_str:

push ax

push cx

push dx

push bx

xor cx,cx ;обнуление CX

mov bx,10 ;В BX делитель

wtuds\_lp1: ;Цикл получения остатков от деления

xor dx,dx ;Обнуление старшей части двойного слова

div bx ;Деление AX=(DX:AX)/BX, остаток в DX

add dl,'0' ;Преобразование остатка в кот символа

push dx ;Сохранение в стеке

inc cx ;Увеличение счетчика символов

test ax,ax ;проверка AX

jnz wtuds\_lp1 ;Переход к началу цикла, если частное не 0.

wtuds\_lp2: ;Цикл извлечения символов из стека

pop dx ; Восстановление символа из стека

mov [di], dl ; Сохранение символа в буфере

inc di ; Инкремент адреса буфера

loop wtuds\_lp2 ; Команда цикла

pop bx

pop dx

pop cx

pop ax

ret ; Возвращаемся к коду вызова функции (call) из основного кода

word\_to\_sdec\_str:

push ax

test ax,ax ;проверка знака AX

jns wtsds\_no\_sign ;если >= 0, преобразуем как беззнаковое

mov byte[di],'-' ;Добавление знака в начало строки

inc di ;увеличиваем DI

neg ax ;изменяем знак значения AX

wtsds\_no\_sign:

call word\_to\_udec\_str ; преобразование беззнакового значения

pop ax

ret ; Возвращаемся к коду вызова функции (call) из основного кода

print\_word\_sdec:

push di

mov di,buffer ;DI = адрес буфера

push di ;сохранение DI в стеке

call word\_to\_sdec\_str ;преобразование слова в AX в строку

mov byte[di],'$' ;Добавление символа конца строки

pop di ;DI = адрес начала строки

call print\_str ;вывод строки на консоль

pop di

ret ; Возвращаемся к коду вызова функции (call) из основного кода

print\_str:

push ax

mov ah,9 ; Функция DOS 09h - вывод строки

xchg dx,di ; Обмен значениями DX и DI

int 21h ; Обращение к функции DOS

xchg dx,di ; Обмен значениями DX и DI

pop ax

ret ; Возвращаемся к коду вызова функции (call) из основного кода

print\_endline:

push di ; заносим в стек значение регистра di

mov di, endline ; DI = Адрес строки CR, LF

call print\_str ; Вывод строки на консоль

pop di ; извлекаем значение регистра

ret ; Возвращаемся к коду вызова функции (call) из основного кода

print\_word\_udec:

push di

mov di, buffer ; DI = Адрес буфера

push di ; Сохранение DI в стеке

call word\_to\_udec\_str ;Преобразование слова в AX в строку

mov byte[di],'$' ; Добавление символа конца строки

pop di ;DI = адрес начала строки

call print\_str ;Вывод строки на консоль

pop di

ret ; Возвращаемся к коду вызова функции (call) из основного кода

переменные:

s\_in\_n db 'Input n: $'

s\_in\_p db 'Input p: $'

s\_in\_a db 'Input a: $'

s\_out db 'Result: $'

s\_c db 'Check: $'

s\_pak db 'Press any key...$'

endline db 13,10,'$'

buffer rb 7

res dw 20 dup(0)

n dw ?