CODE:

```
.model small
.data
  arr1    db 9,5,0,3,2,3,1,7,1,4
  OddArr  db 10 dup(?)
  EvenArr db 10 dup(?)
  cnt_0   db 0
  cnt_e   db 0

.code
  mov ax,@data            ;Initialize data segment
  mov ds,ax

 mov cx,10                ;Initialize word counter
 mov si,offset arr1       ;Initialize memory counter
 mov di,offset Oddarr
 mov bx,offset EvenArr
up:mov al,[si]
ror al,1                  ;rotate number by 1 bit towards left
jnc dn                    ;check number odd or even
rol al,1                  ;if odd then restore the number
mov [di],al               ;write number to destination array
inc di
  inc cnt_0
  jmp dn1
dn:rol al,1
  mov [bx],al
  inc bx
  inc cnt_e
dn1:inc si
    loop up

    mov ch,0              ;initialize pass counter
    mov cl,cnt_0
    mov ah,cnt_0
    dec ah
next: mov bl,ah           ;initialize comparizon counter
      mov si,offset OddArr ;initialize memory counter
up1:  mov al,[si]
      cmp al,[si+1]       ;compare number with next number
      jc dn2              ;if number>next number then go to dn
      xchg al,[si+1]
      xchg al,[si]
```

```
dn2:  inc si              ;initialize memory counter by 1
      dec bl              ;decrement comparison counter by 1
      jnz up1             ;if comparison is not zero then go to dn
      loop next

      mov ch,0
      mov cl,cnt_0        ;Initialize pass counter
      dec ah

next1:mov bl,ah
      mov si,offset EvenArr
up2:  mov al,[si]
      cmp al,[si+1]
      jc dn3
      xchg al,[si+1]
      xchg al,[si]        ;exchange number with next number
dn3:  inc si              ;increment memory pointer by 1
      dec bl              ;decrement comparison counter by
      jnz up2             ;if comparison is not zero then go to dn
      loop next1          ;decrement pass counter by 1,if not zero then goto jnz
      mov ch,0
      mov cl,cnt_e
      dec ah
      mov ch,0
      mov cl,cnt_0        ;Initialize pass counter
      dec ah

end                       ;end segment
ends                      ;end of ALP
```