

;
;Ucitava se niz N1. Ukoliko je moguće, formirati niz N2 od članova
niza N1 tako da se nigde ne pojavljuju dva parna ili dva neparna
člana susedna

```
%include "asm_io.inc"  
segment .data
```

```
poruka1 db "Unesite broj N: ",0  
poruka2 db "Unesite celobrojnu vrednost: ",0  
poruka3 db "Niје moguće formirati niz2!",0  
poruka4 db "Niz N1: ",0  
poruka5 db "Niz N2: ",0  
const dd 2  
br_parnih dd 0
```

```
segment .bss
```

```
niz resd 100  
niz2 resd 100  
N resd 1  
x resd 1  
esi_num resd 1
```

```
segment .text  
    global  _asm_main  
_asm_main:  
    enter   0,0                ; setup routine  
    pusha  
  
    mov eax, poruka1  
    call print_string  
    call read_int  
    mov [N], eax ; ukupan broj članova  
    mov ecx, 0 ; brojac trenutnih članova niza  
    mov esi, 0  
  
formiranje_niza:  
    cmp ecx, [N]  
    je izracunavanje_potrebnih_parametara  
    mov eax, poruka2  
    call print_string  
    call read_int  
    mov [niz+esi], eax  
    inc ecx  
    add esi, 4  
    jmp formiranje_niza
```

```
izracunavanje_potrebnih_parametara:
    mov eax, esi
    mov [esi_num], eax

broj_parnih:
    cmp esi, 0
    je obrada
    sub esi, 4
    mov eax, [niz+esi]
    mov ebx, 2
    mov edx, 0
    div ebx
    cmp edx, 0
    jne broj_parnih
    inc dword [br_parnih]
    jmp broj_parnih

obrada:
    mov eax, ecx
    sub eax, [br_parnih] ; ostaje broj neparnih
    sub eax, [br_parnih] ; ostaje razlika neparnih i parnih
    mul eax ; kvadiramo razliku. Dozvoljena razlika je -
1,0,+1.
    cmp eax, 1
    ja nije_moguce
    jmp niz2_formiranje

nije_moguce:
    mov eax, poruka3
    call print_string
    call print_nl
    jmp kraj

niz2_formiranje:
    mov eax, ecx
    sub eax, [br_parnih] ; ostaje broj neparnih
    sub eax, [br_parnih] ; ostaje razlika neparnih i parnih
    ;dump_regs 2
    shl eax, 1 ; ispitujemo znak razlike broja parnih i
neparnih kako bi odredili koji element treba biti prvi u nizu N2
    jc prvi_paran
prvi_neparan:
    mov esi, 0
    mov ebx, 0 ; brojac za neparne
    mov ecx, 4 ; brojac za parne
    jmp dalje
prvi_paran:
    mov esi, 0
    mov ebx, 4 ; brojac za neparne
    mov ecx, 0 ; brojac za parne
```

dalje:

```
    mov eax, [esi_num]
    cmp eax, esi
    je ispisi_niz
    mov eax, [niz+esi]
    mov edx, 0
    div dword [const]
    cmp edx, 0
    je paran
    mov eax, [niz+esi]
    mov [niz2+ebx], eax
    add ebx, 8
    add esi, 4
    jmp dalje
```

paran:

```
    mov eax, [niz+esi]
    mov [niz2+ecx], eax
    add ecx, 8
    add esi, 4
    jmp dalje
```

ispisi_niz:

```
    mov esi, 0
    mov eax, poruka5
    call print_string
    call print_nl
```

ispis:

```
    mov eax, [esi_num]
    sub eax, 4
    cmp eax, esi
    jb kraj
    mov eax, [niz2+esi]
    call print_int
    call print_nl
    add esi, 4
    jmp ispis
```

kraj:

```
    popa
    mov     eax, 0                ; return back to C
    leave
    ret
```