

Задача 31: Програма виводить парні числа в інтервалі [1, 19). Зробіть, щоб виводила їх в інтервалі [4-19).

- 1) Скачав і створив копію ELF
- 2) Запустив ELF визначив що воно принтує починаючи з 0 а не з 1 (оскільки парні то навіть з 2)

```
fenix@charli ~/pr/subj/POK/lab_3_asm/hack_the_exe_task2$ ls
new_prg_31.x  prg_31.x
fenix@charli ~/pr/subj/POK/lab_3_asm/hack_the_exe_task2$ ./prg_31.x
0
2
4
6
8
10
12
14
16
18
```

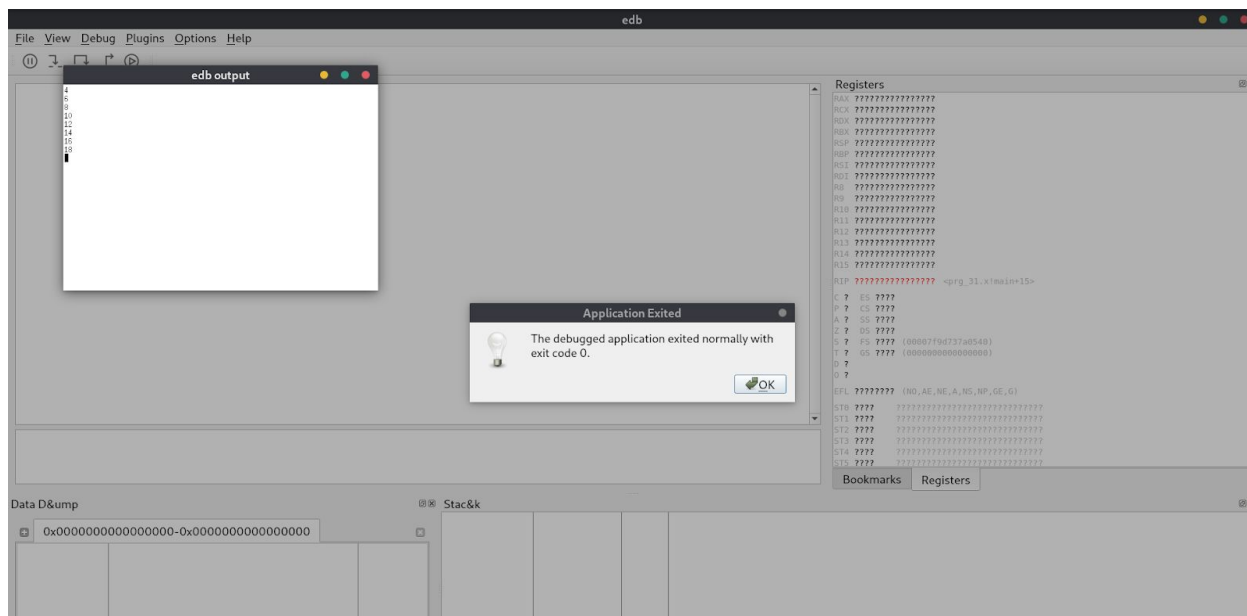
Деасемблював за допомогою 'plasma' і проаналізував як працює код.

```
fenix@charli ~/pr/subj/POK/lab_3_asm/hack_the_exe_task2$ plasma prg_31.x
function main (.text) {
    0x4004d0: push rbp
    0x4004d1: rbp = rsp
    0x4004d4: rsp -= 16
    0x4004d8: *(rbp - 4) = 0
    0x4004df: *(rbp - 8) = 0
    loop {
        loop_0x4004e6:
            # 0x4004e6: cmp dword ptr [rbp - 8], 0x14
            # 0x4004ea: jge 0x400536
            if (*(rbp - 8) >= 20) goto ret_0x400536
            0x4004f0: eax = 2
            0x4004f5: ecx = *(rbp - 8)
            0x4004f8: *(rbp - 12) = ecx
            0x4004fb: eax = ecx
            0x4004fd: cdq
            0x4004fe: ecx = *(rbp - 12)
            0x400501: eax = edx:eax / ecx; edx = edx:eax % ecx
            # 0x400503: cmp edx, 0
            # 0x400506: jne 0x400523
            if (edx == 0) {
                0x40050c: movabs rdi, 0x4005c4 "%i\n"
                0x400516: esi = *(rbp - 8)
                0x400519: al = '\0'
                0x40051b: call printf
                0x400520: *(rbp - 16) = eax
            }
            0x400523: jmp 0x400528
            0x400528: eax = *(rbp - 8)
            0x40052b: eax += 1
            0x40052e: *(rbp - 8) = eax
            0x400531: jmp loop_0x4004e6
    } ; loop_0x4004e6

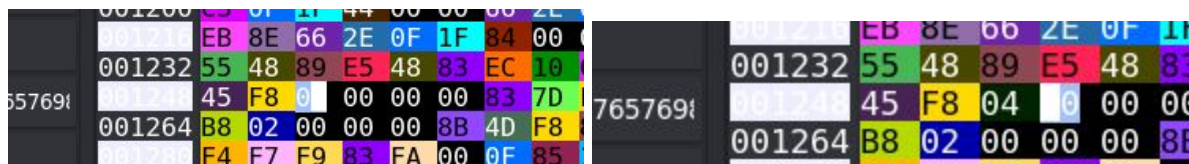
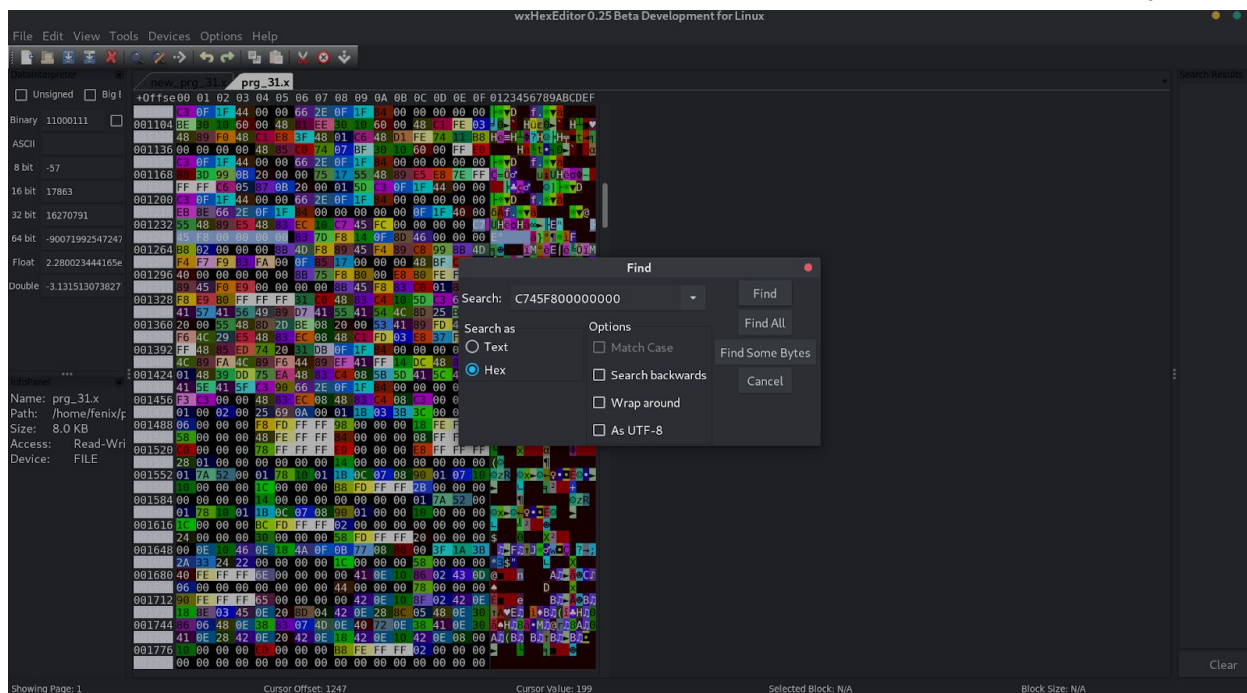
    ret_0x400536:
    0x400536: eax = 0
    0x400538: rsp += 16
    0x40053c: pop rbp
    0x40053d: ret
}
```

Зробив висновок, що виводиться значення з пам'яті за адресою `rbp-8` якщо воно парне. Цикл закінчується коли значення за адресою `rbp-8` ≥ 20 . Отже, якщо змінити початкове значення, яке присвоюється комірці за адресою `rbp-8` на 4, то отримаємо бажаний результат.

Запустивши програму до кінця отримав підтвердження, що програма працює так як треба.



5) Тоді зайшов в бінарний редактор wxHexEditor і задавши в пошук достатньо довгу послідовність байт (с7 45 f8 00 00 00 00) знайшов необхідне місце для редагування.



Змінив один байт (номер якого 19972). Зберіг новоутворений ELF як new_prg_31.x і перевірів справність.

```
fenix@charli > ~/pr/subj/P0K/lab_3_asm/hack_the_exe_task2 > master > ./prg_31.x
0
2
4
6
8
10
12
14
16
18
fenix@charli > ~/pr/subj/P0K/lab_3_asm/hack_the_exe_task2 > master > ./new_prg_31.x
4
6
8
10
12
14
16
18
```