

№3 hocus-pocus

В общем, objdump выдает мне деассемблированный бинарник, который виден в виде ассемблера, а работу программы через код ASM сложно понять, поэтому запустим через интернет-деассемблер, который даст мне псевдокод.

```
47  int v3; // [rsp+0h] [rbp-10h] BYREF
48  int v4[4]; // [rsp+4h] [rbp-14h] BYREF
49  int v5; // [rsp+14h] [rbp-4h]
50
51  v4[0] = a1;
52  v3 = a2;
53  v5 = a1 * a2;
54  if ( a1 < a2 )
55    hocus_pocus(v4, &v3);
56  while ( v3 )
57  {
58    v4[0] %= v3;
59    hocus_pocus(v4, &v3);
60  }
```

Равносильные два кода

почти что

В

общем,

поиск

```
local_c = param_2 * param_1;
local_20 = param_2;
local_1c[0] = param_1;
if (param_1 < param_2) {
    hocus_pocus(local_1c,&local_20);
}
while (local_20 != 0) {
    local_1c[0] = local_1c[0] % local_20;
    hocus_pocus(local_1c,&local_20);
}

void hocus_pocus(undefined4 *param_1,undefined4 *param_2)
{
    undefined4 uVar1;

    uVar1 = *param_1;
    *param_1 = *param_2;
    *param_2 = uVar1;
    return;
}
```

это

наименьшего общего кратного.

В итоге, подбираем для следующих пунктов просто нужный ввод, чтобы получить определенные числа.

Например, для вывода 42 возьмем 42 и 1

Для вывода 0 возьмем 0 и другое число не равное 0

Для вывода ошибки возьмем два нуля

Для вывода int_min возьмем это число (-2147483648) и 1

Дальше нужно написать эквивалентный код (с той же ошибкой на нуле). Код на C лежит в папке в файле code.c (почти равносильный тому, что мне предлагает подобие IDA)