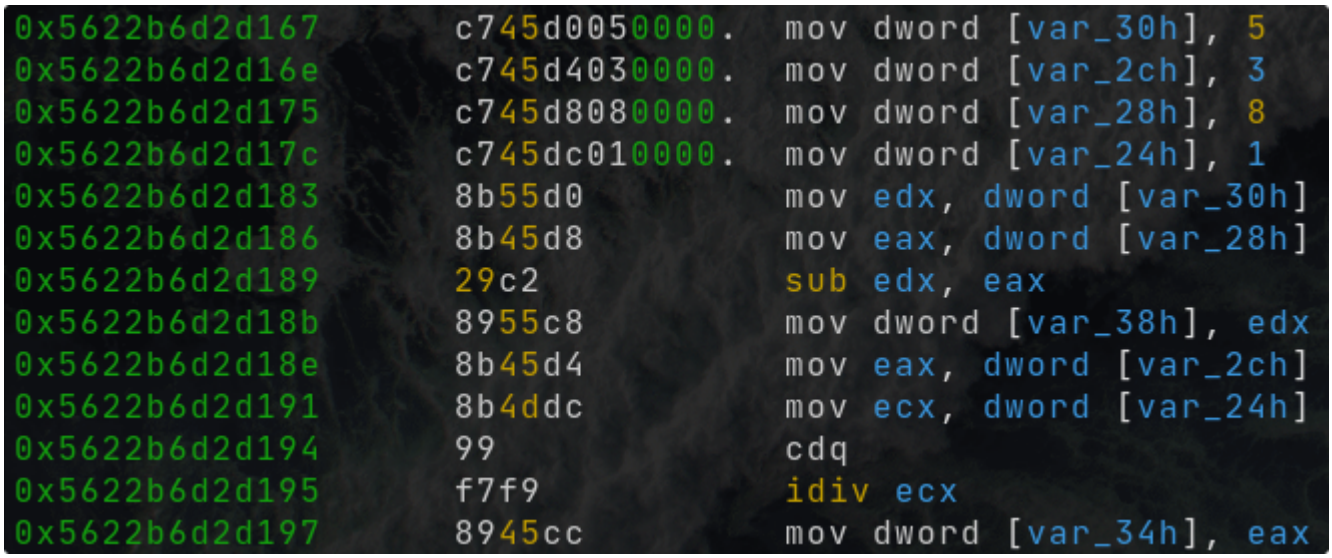
Стан регістрів на кожному пункті

1. Массив елементів та дії віднімання та ділення на них

За цей пункт відповідає код на C:

```c
int arr[] = { 5, 3, 8, 1 };
int res1 = arr[0] - arr[2];
int res2 = arr[1] / arr[3];
```

Йому відповідає така діляка дизассембльованого коду



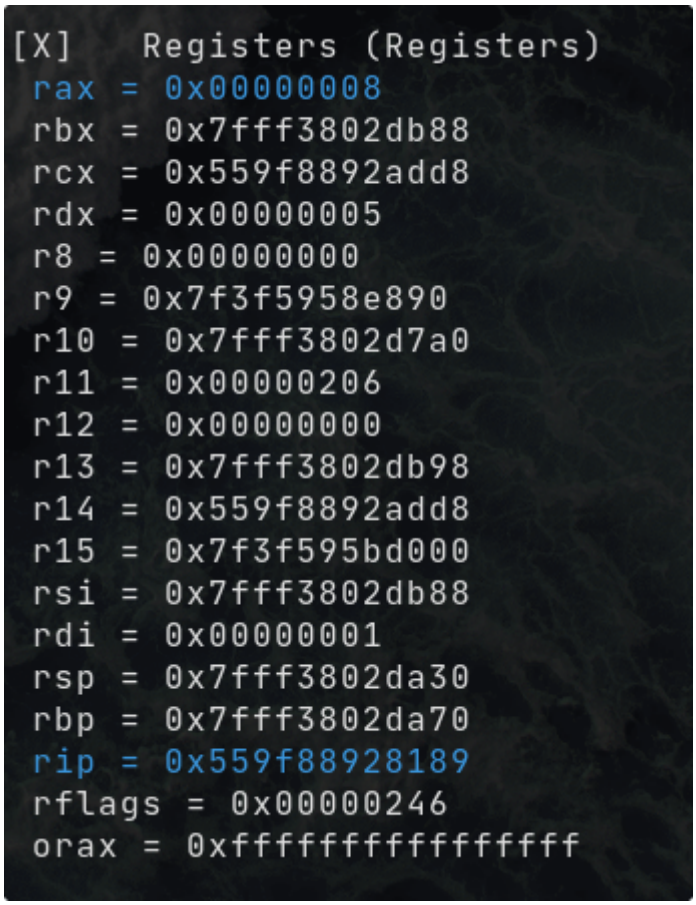
Як елементи масиву дебагер показує локальні змінні `var_30h`, `var_2ch`, `var_28h`, `var_24h`, що відповідають елементам 0, 1, 2 та 3 массиву.

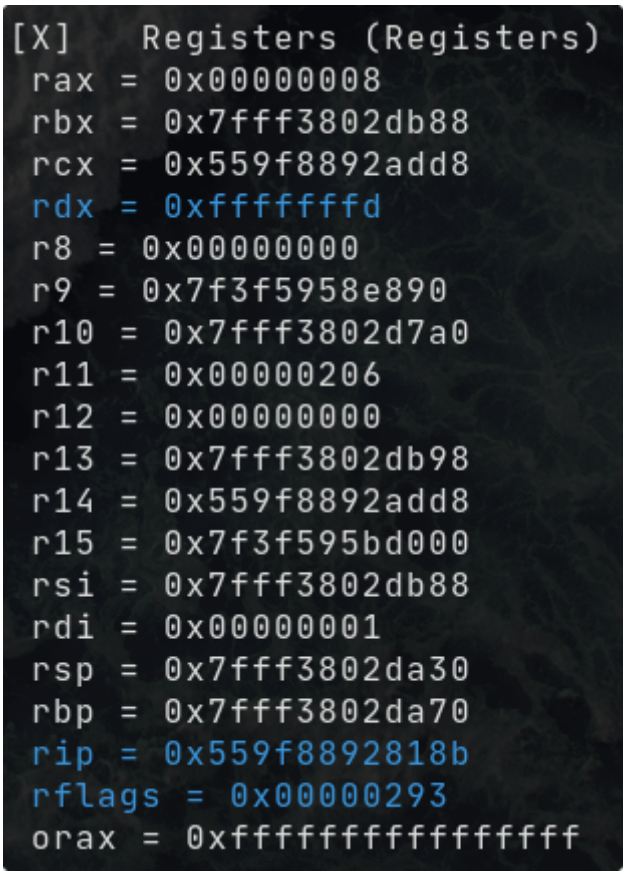
Також ми бачимо:

- запис (адреси `67`, `6e`, `75`, `7c`) захардкожених даних у елементи масиву
- процес віднімання та запису результату виконання опкоду `sub` з регістру `edx` у область змінної `var_38h` по адресам `83`, `86`, `89`, `8b`. Для віднімання використовуються регістри `edx` (data register) та `eax` (primary accumulator).
- за ділення відповідають команди за адресами `8e`, `91`, `94`, `95`. `idiv` записує результат у регістри `eax` та `edx`

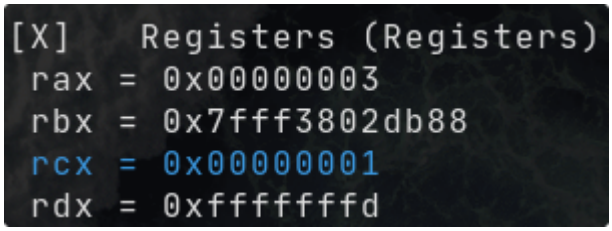
Значення у регістрах:

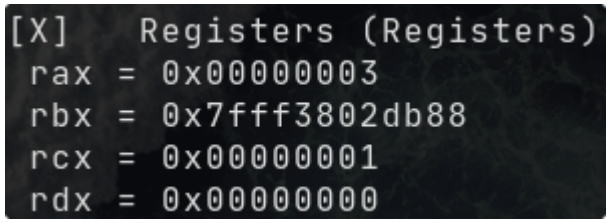
- на момент запису нульового та другого елементів у `edx` та `eax` перед відніманням та після операції віднімання результат у регістрі `rdx`





- запис даних у регістри `eax` та `ecx` для ділення, після ділення





2. Блок `if`

```
if (arr[1] == 42) {
  arr[1] = 24;
}
```

Йому відповідає

```
    0x5622b6d2d19a      8b45d4          mov eax, dword [var_2ch]
    0x5622b6d2d19d      83f82a          cmp eax, 0x2a               ; 42
┌<  0x5622b6d2d1a0      7507            jne 0x5622b6d2d1a9
│   0x5622b6d2d1a2      c745d4180000.   mov dword [var_2ch], 0x18   ; 24
```

Де записується значення першого елементу масиву у регістр `eax` та далі порівнюємо це значення із значенням `0x2a` (42). Якщо ж два значення не рівні, то виконуємо перехід на адресу `0x5622b6d2d1a9`, інакше записуємо у перший елемент масиву `0x18` (24).

```
[X]   Registers (Registers)
 rax = 0x00000003
 rbx = 0x7fffea9c9988
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fc8393b9890
 r10 = 0x7fffea9c95a0
 r11 = 0x00000206
 r12 = 0x00000000
 r13 = 0x7fffea9c9998
 r14 = 0x5622b6d2fdd8
 r15 = 0x7fc8393e8000
 rsi = 0x7fffea9c9988
 rdi = 0x00000001
 rsp = 0x7fffea9c9830
 rbp = 0x7fffea9c9870
 rip = 0x5622b6d2d19a
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

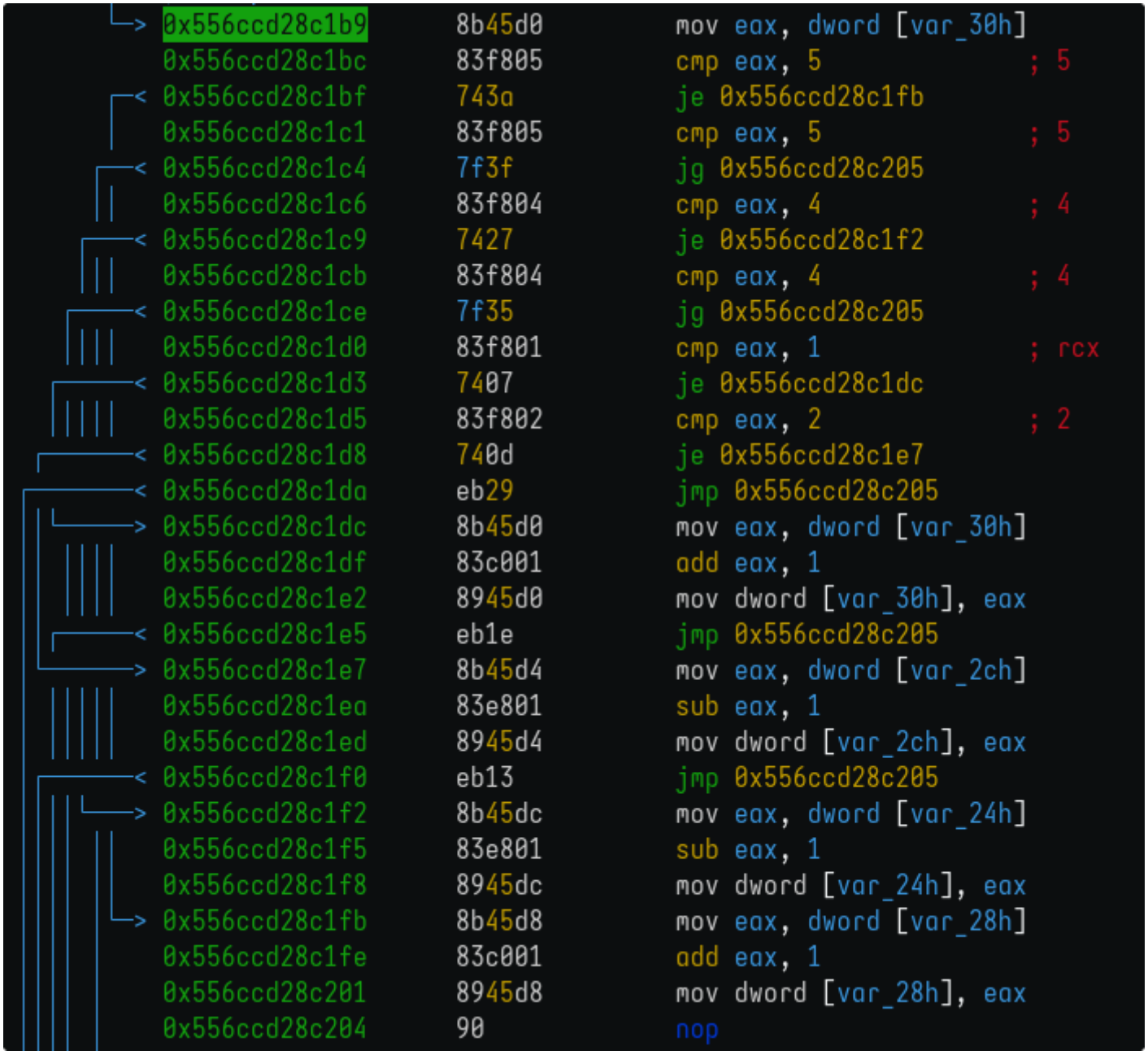
```
[X]   Registers (Registers)
 rax = 0x00000003
 rbx = 0x7fffea9c9988
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fc8393b9890
 r10 = 0x7fffea9c95a0
 r11 = 0x00000206
 r12 = 0x00000000
 r13 = 0x7fffea9c9998
 r14 = 0x5622b6d2fdd8
 r15 = 0x7fc8393e8000
 rsi = 0x7fffea9c9988
 rdi = 0x00000001
 rsp = 0x7fffea9c9830
 rbp = 0x7fffea9c9870
 rip = 0x5622b6d2d19d
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000003
 rbx = 0x7fffea9c9988
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fc8393b9890
 r10 = 0x7fffea9c95a0
 r11 = 0x00000206
 r12 = 0x00000000
 r13 = 0x7fffea9c9998
 r14 = 0x5622b6d2fdd8
 r15 = 0x7fc8393e8000
 rsi = 0x7fffea9c9988
 rdi = 0x00000001
 rsp = 0x7fffea9c9830
 rbp = 0x7fffea9c9870
 rip = 0x5622b6d2d1a0
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000003
 rbx = 0x7ffcfeb70e98
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7f9ca9334890
 r10 = 0x7ffcfeb70ab0
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffcfeb70ea8
 r14 = 0x560fb9686dd8
 r15 = 0x7f9ca9363000
 rsi = 0x7ffcfeb70e98
 rdi = 0x00000001
 rsp = 0x7ffcfeb70d40
 rbp = 0x7ffcfeb70d80
 rip = 0x560fb96841a9
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

3. `switch`

```
switch (arr[0]) {
  case 1:
    arr[0]++;
    break;
  case 2:
    arr[1]--;
    break;
  case 4:
    arr[3]--;
  case 5:
    arr[2]++;
    break;
}
```



Значення у регістрах:

```
[X]   Registers (Registers)
 rax = 0x00000003
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c1b9
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c1bc
 rflags = 0x00000293
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c1bf
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c1fb
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000008
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c1fe
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000009
 rbx = 0x7fff95c83178
 rcx = 0x00000001
 rdx = 0x00000000
 r8 = 0x00000000
 r9 = 0x7fe66530b890
 r10 = 0x7fff95c82d90
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7fff95c83188
 r14 = 0x556ccd28edd8
 r15 = 0x7fe66533a000
 rsi = 0x7fff95c83178
 rdi = 0x00000001
 rsp = 0x7fff95c83020
 rbp = 0x7fff95c83060
 rip = 0x556ccd28c201
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

4. Global variable

Вивід значення змінної (виклик `printf`)

```
; CODE XREFS from main @ 0x556ccd28c1da, 0x556ccd28c1e5, 0x556ccd28c1f0
0x556ccd28c205      8b05152e0000    mov eax, dword [obj.global_variable]    ; [0x556ccd28f020:4]=0x6ed6
0x556ccd28c20b      89c6            mov esi, eax
0x556ccd28c20d      488d05f00d00.   lea rax, [0x556ccd28d004]    ; "%d\n"
0x556ccd28c214      4889c7          mov rdi, rax
0x556ccd28c217      b800000000      mov eax, 0
0x556ccd28c21c      e81ffeffff      call sym.imp.printf     ;[1] ; int printf(const char *format)
```

Значення глобальної змінної знаходиться за адресою `0x556ccd28f020`

```
[X]   Hexdump (Hexdump)                                                    [Cache] Off
 - offset -       0 1  2 3  4 5  6 7  8 9  A B  C 0123456789ABC  comment
 0x556ccd28f020  d66e 0000 0000 0000 0000 0000 00 .n...........  ; obj.global_variable  ;
 0x556ccd28f02d  0000 0000 0000 0000 0000 0000 00 .............
 0x556ccd28f03a  0000 0000 0000 0000 0000 0000 00 .............
```

5. Цикл `while`

```
// 5. while cycle
while (arr[0] > 0) {
  arr[0]--;
}
```

Перевіряється умова циклу

```
[X]   Registers (Registers)
 rax = 0x00000006
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3221
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000006
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b322c
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b322f
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3231
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

Перейшли у тіло циклу

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3223
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000005
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3226
 rflags = 0x00000206
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000004
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3229
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

Та знову перейшли на область перевірки умови

```
[X]   Registers (Registers)
 rax = 0x00000004
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b322c
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

6. Реалізована функція `is_even`, що перевіряє `int` число на подільність на 2.

Її виклик в `main`

```
0x55d89e9b3233      8b45dc          mov eax, dword [var_24h]
0x55d89e9b3236      89c7            mov edi, eax
0x55d89e9b3238      e80cffffff      call sym.is_even        ;[2]
```

Значення регістрів при виклику

```
[X]   Registers (Registers)
 rax = 0x00000000
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3233
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x7ffeb1dbe660
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3236
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebc0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3238
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

Функція `is_even`

Та значення регістрів при її виконанні

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb8
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b3149
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbec00
 rip = 0x55d89e9b314a
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b314d
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b3150
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b3153
 rflags = 0x00000246
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b3156
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000001
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b3158
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

```
[X]   Registers (Registers)
 rax = 0x00000000
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b315b
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

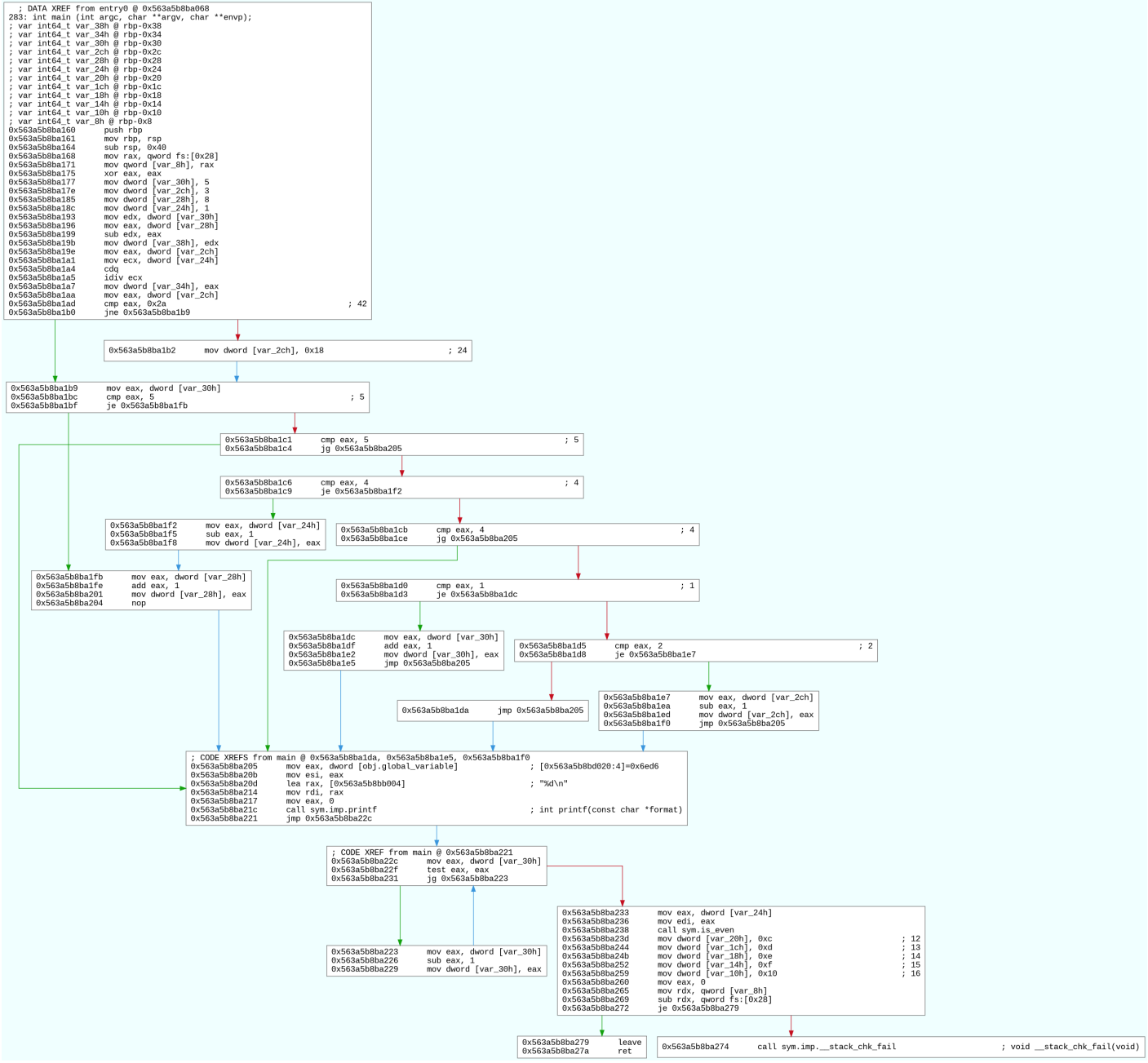
```
[X]   Registers (Registers)
 rax = 0x00000000
 rbx = 0x7ffeb1dbed18
 rcx = 0x00000000
 rdx = 0x00000000
 r8 = 0x00000400
 r9 = 0x00000064
 r10 = 0x7fb95a141c00
 r11 = 0x00000202
 r12 = 0x00000000
 r13 = 0x7ffeb1dbed28
 r14 = 0x55d89e9b5dd8
 r15 = 0x7fb95a1b3000
 rsi = 0x55d89f5112a0
 rdi = 0x00000001
 rsp = 0x7ffeb1dbebb0
 rbp = 0x7ffeb1dbebb0
 rip = 0x55d89e9b315e
 rflags = 0x00000202
 orax = 0xffffffffffffffff
```

7. Список зі значеннями

```
// 7. array with specific elements
int arr2[] = { 12, 13, 14, 15, 16 };
```

```
0x55d89e9b323d      c745e00c0000.   mov dword [var_20h], 0xc    ; 12
0x55d89e9b3244      c745e40d0000.   mov dword [var_1ch], 0xd    ; 13
0x55d89e9b324b      c745e80e0000.   mov dword [var_18h], 0xe    ; 14
0x55d89e9b3252      c745ec0f0000.   mov dword [var_14h], 0xf    ; 15
0x55d89e9b3259      c745f0100000.   mov dword [var_10h], 0x10    ; 16
```

Дерево програми (функція main)



Лістинг програми на C

```c
#include <stdio.h>

// 4. global variable
int global_variable = 28374;

// 6. own function
int is_even(int num) {
  return num % 2 == 0;
}

int main() {
  // 1. array elements and actions (division, substraction)
  int arr[] = { 5, 3, 8, 1 };
  int res1 = arr[0] - arr[2]; int res2 = arr[1] / arr[3];

  // 2. if statement
```

```c
  if (arr[1] == 42) {
    arr[1] = 24;
  }

  // 3. switch on 4 cases
  switch (arr[0]) {
    case 1:
      arr[0]++;
      break;
    case 2:
      arr[1]--;
      break;
    case 4:
      arr[3]--;
    case 5:
      arr[2]++;
      break;
  }

  // 4. print global variable
  printf("%d\n", global_variable);

  // 5. while cycle
  while (arr[0] > 0) {
    arr[0]--;
  }

  // 6. function call
  is_even(arr[3]);

  // 7. array with specific elements
  int arr2[] = { 12, 13, 14, 15, 16 };

  return 0;
}
```