Level 0x04

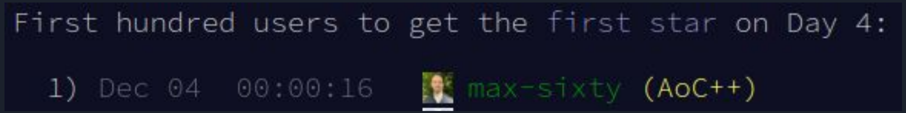
More C

Topics

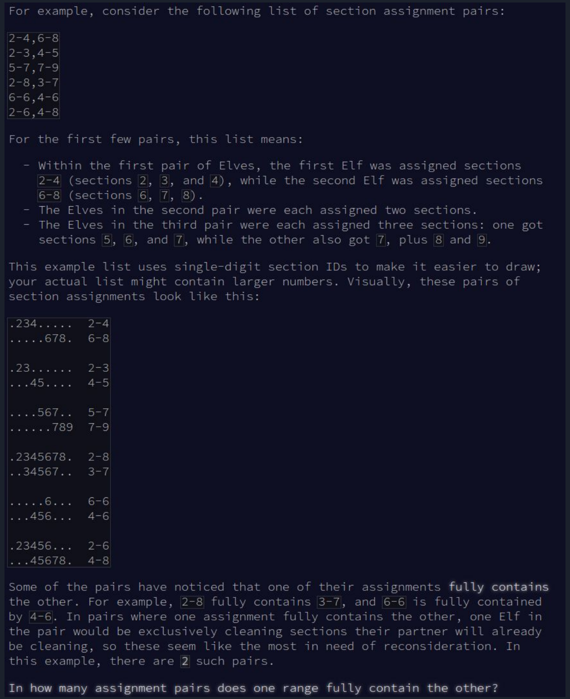
- Events
- Hacker History
- C Stuff
  - Compiling
  - Printf
  - Operators (shifts, xor)

A.I. Wars

- Advent of Code, Day 4
- Challenge completed in 16 seconds
  - Chat GPT
  - https://github.com/max-sixty/aoc-gpt



```
First hundred users to get the first star on Day 4:

  1) Dec 04  00:00:16   max-sixty (AoC++)
```



For example, consider the following list of section assignment pairs:

```
2-4,6-8
2-3,4-5
5-7,7-9
2-8,3-7
6-6,4-6
2-6,4-8
```

For the first few pairs, this list means:

- Within the first pair of Elves, the first Elf was assigned sections 2-4 (sections 2, 3, and 4), while the second Elf was assigned sections 6-8 (sections 6, 7, 8).
- The Elves in the second pair were each assigned two sections.
- The Elves in the third pair were each assigned three sections: one got sections 5, 6, and 7, while the other also got 7, plus 8 and 9.

This example list uses single-digit section IDs to make it easier to draw; your actual list might contain larger numbers. Visually, these pairs of section assignments look like this:

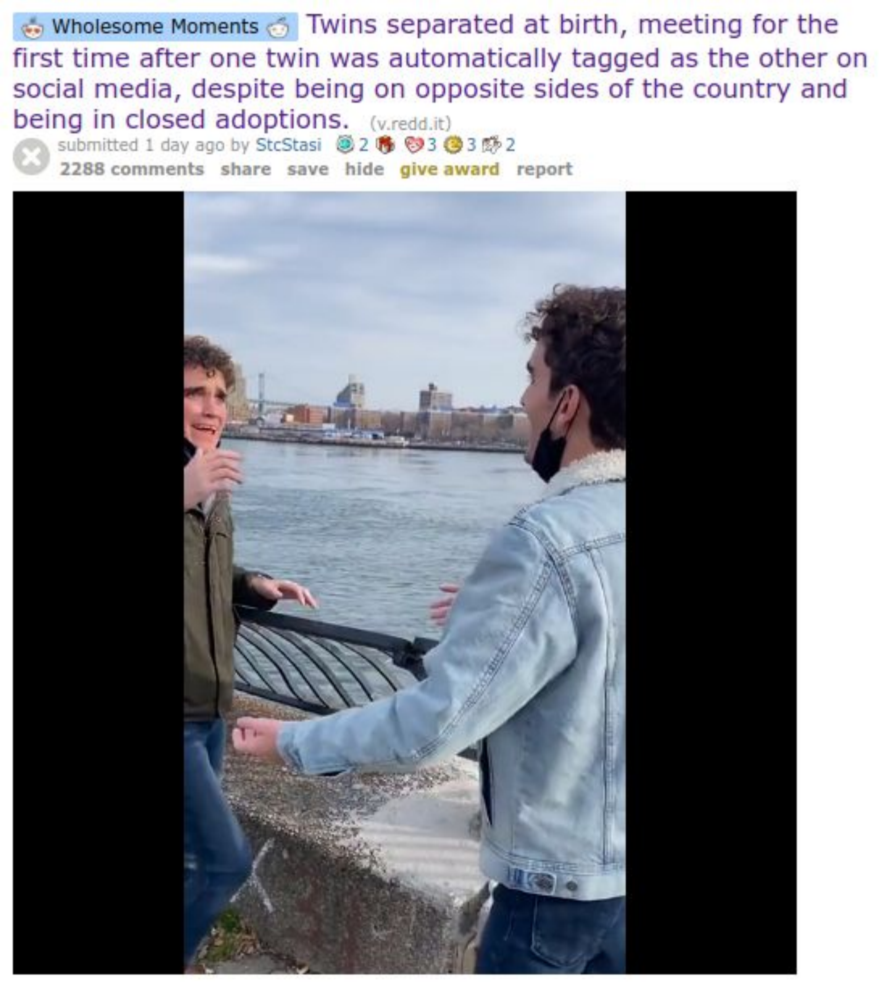
```
.234.....  2-4
.....678.  6-8

.23......  2-3
...45....  4-5

....567..  5-7
......789  7-9

.2345678.  2-8
..34567..  3-7

.....6...  6-6
...456...  4-6

.23456...  2-6
...45678.  4-8
```

Some of the pairs have noticed that one of their assignments **fully contains** the other. For example, 2-8 fully contains 3-7, and 6-6 is fully contained by 4-6. In pairs where one assignment fully contains the other, one Elf in the pair would be exclusively cleaning sections their partner will already be cleaning, so these seem like the most in need of reconsideration. In this example, there are 2 such pairs.

In how many assignment pairs does one range fully contain the other?

Facial Recognition

- Front page of Reddit
  - It's possibly a fake story
- Other impacts?
  - Undercover / secret agents
  - Witness Protection Program
  - China's "Social Credit System"
  - Trustworthiness of people without social media history



Wholesome Moments Twins separated at birth, meeting for the first time after one twin was automatically tagged as the other on social media, despite being on opposite sides of the country and being in closed adoptions. (v.redd.it)

submitted 1 day ago by StcStasi 2 3 3 2

2288 comments share save hide give award report

Upcoming Events

- CyberQuest (Lockheed Martin, Orlando)
    - Saturday, March 4th, 3 hour competition
    - Registration ENDS TODAY!
    - High school
    - https://cybercodequest.lockheedmartin.com

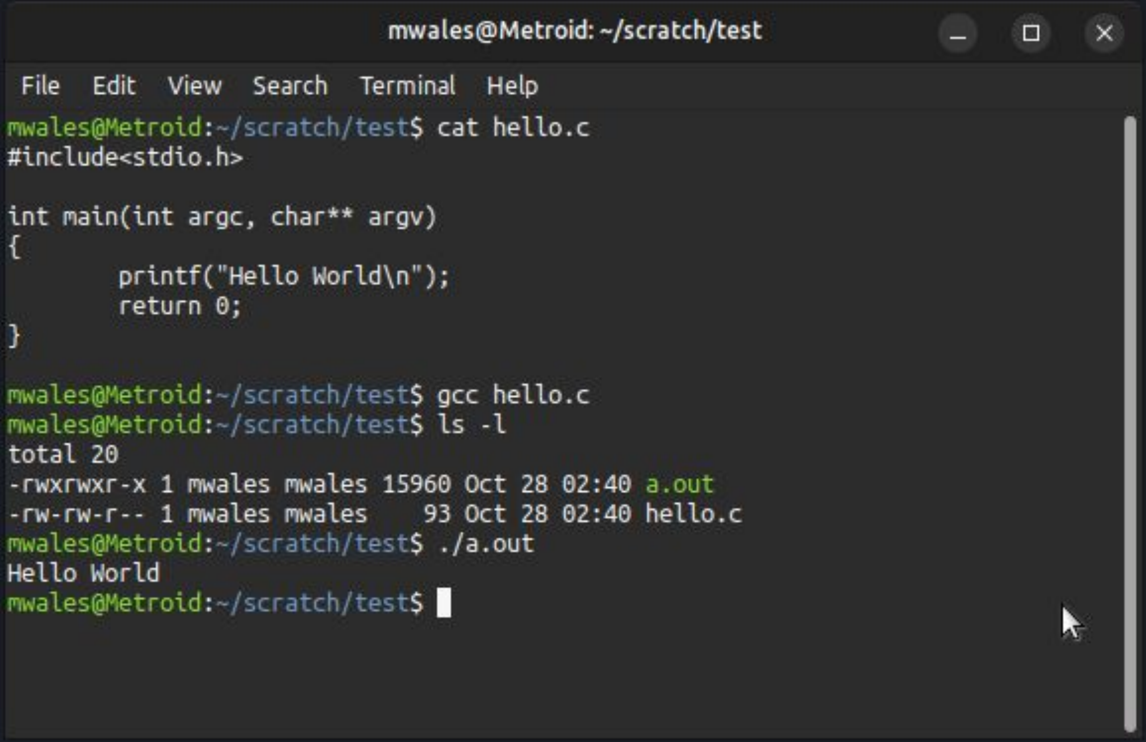
LOCKHEED MARTIN
CYBERQUEST®
COMPETITION

Hello World

- Comments
  - /* orig C multi-line */
  - // C++ style single line
- Single main function
  - You can access command line args
  - Return code to the calling script
    - 0 = success
- printf = print formatted
- Indentation doesn’t matter

```
#include <stdio.h>

/*
  Multi-line comments!

  argc is number of args to program
  argv is list of the args
*/

int main(int argc, char** argv)
{
   // Every C program has a main function
   printf("Hello World\n");
   return 0;
}
```

Compiling C applications


mwales@Metroid: ~/scratch/test
File Edit View Search Terminal Help

```
mwales@Metroid:~/scratch/test$ cat hello.c
#include<stdio.h>

int main(int argc, char** argv)
{
        printf("Hello World\n");
        return 0;
}

mwales@Metroid:~/scratch/test$ gcc hello.c
mwales@Metroid:~/scratch/test$ ls -l
total 20
-rwxrwxr-x 1 mwales mwales 15960 Oct 28 02:40 a.out
-rw-rw-r-- 1 mwales mwales    93 Oct 28 02:40 hello.c
mwales@Metroid:~/scratch/test$ ./a.out
Hello World
mwales@Metroid:~/scratch/test$ 
```

Variables / Integer Types

- Character type + Integer Type
  - char
- Integer types
  - short
  - int
  - long
  - long long
- Floating point type
  - float
  - double
  - long double
- Size Type
  - size_t
- Integer types can also be unsigned

```
// old standard declaration
char smallNum, otherNum;
smallNum = 10;
otherNum = 42;

// typical declaration + assign
int numPoints = 10000;
unsigned long altitude = 5000;

// OK to reassign later
smallNum = 0xb7;  // hex

// doubles are more precise
float pi = 3.14159;
double r = 45.0;

double area = pi * r * r;
```

Printf formatting

- Escape sequences
  - `\n` is a newline
  - `\t` is a tab
- Conversions / extra args
  - `%d` or `%i` for signed integer, `%u` for unsigned integer
  - `%x` or `%X` for hexadecimal numbers
  - `%c` for single characters
  - `%s` for strings
  - `%f` for floating point
- Flags, width, and precision
  - `printf(“Name: %10s  Weight%3.1f\n”, name, weight);`
  - `printf(“32 bit num in hex is 0x%08x\n”, 0xbeef);`
- Not just C uses it…
  - Python: `“format str %s = %d” % (“age”, 12)`
  - Java: `System.out.println(format, args)`

Arithmetic

- Basic operations: add (+), subtract (-), multiply (*), divide (/)
- Modulus operations: % (division remainder of)
- Increment (++) and Decrement (--)
  - Prefix form: int xVal = ++i; // i is incremented, xVal is then set to value of i
  - Postfix form: int xVal = i++; // xVal is set to the value of i, then i is incremented
- Assignment and operations can be combined
  - Operations followed by =
  - varX *= 2; // equivalent to varX = varX * 2
- Bitwise Operators
  - Operates on bits of value (see previous slides)
  - And (&), Or (|), Xor (^)
  - Shift operation
    - Left shift (<<) and right shift (>>)
    - 0b0001110 << 2 is equivalent to 0b0111000

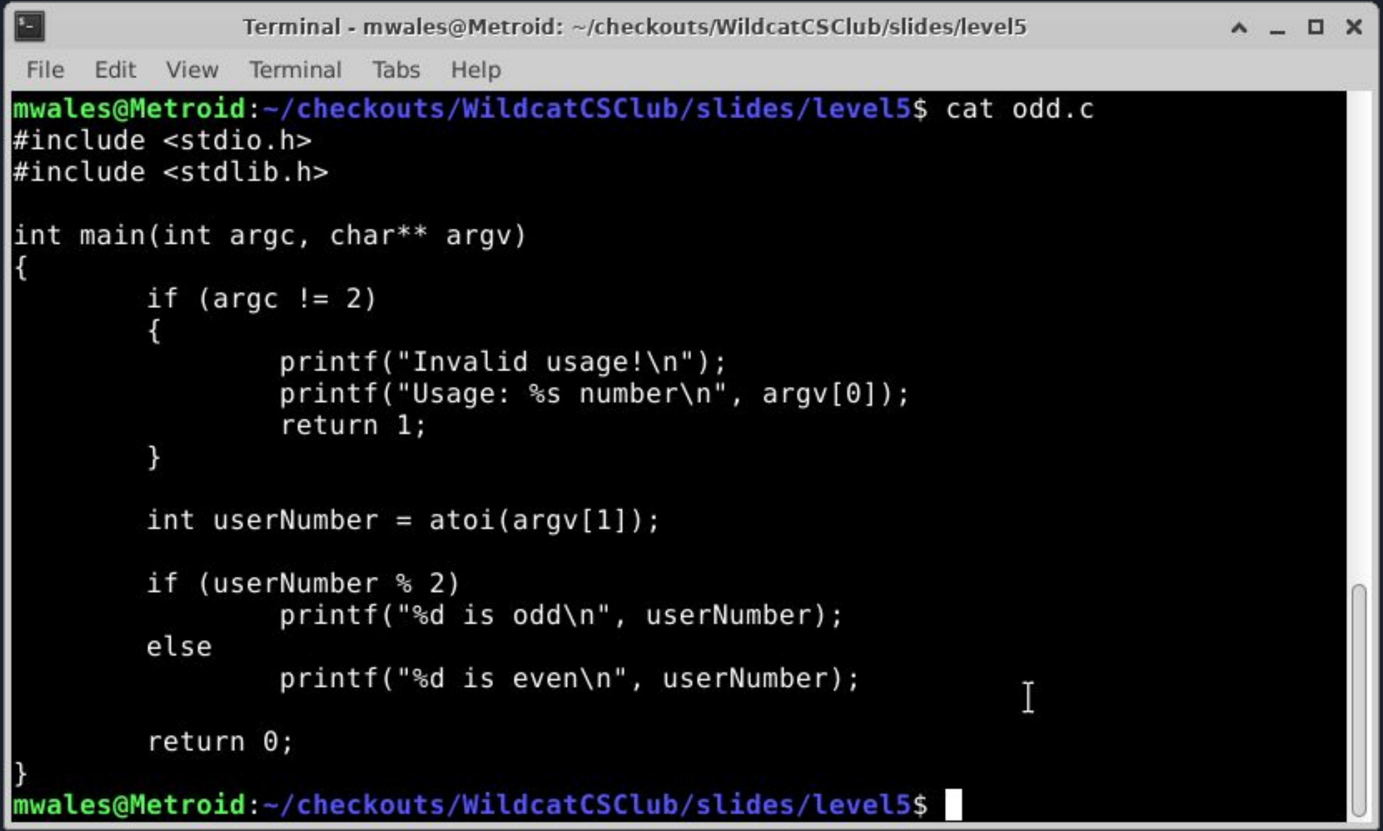
Branching

- Simple branch

```
if (test_expression)
    printf("Test must be true!\n");
```

- Code blocks
  - Can write multiple statements where a bare if statement only allows one statement
  - Variables declared inside code block, only exist in code block
  - Surrounded by curly braces
- Else statement
  - Executes if test expression is false
  - There is no elif like python (in C we would write else if (another test)
- The test expression
  - Can be 0 or non-zero, True or False
  - Can use boolean && or || operators to make complex conditions
- Be careful about semicolon placement!!

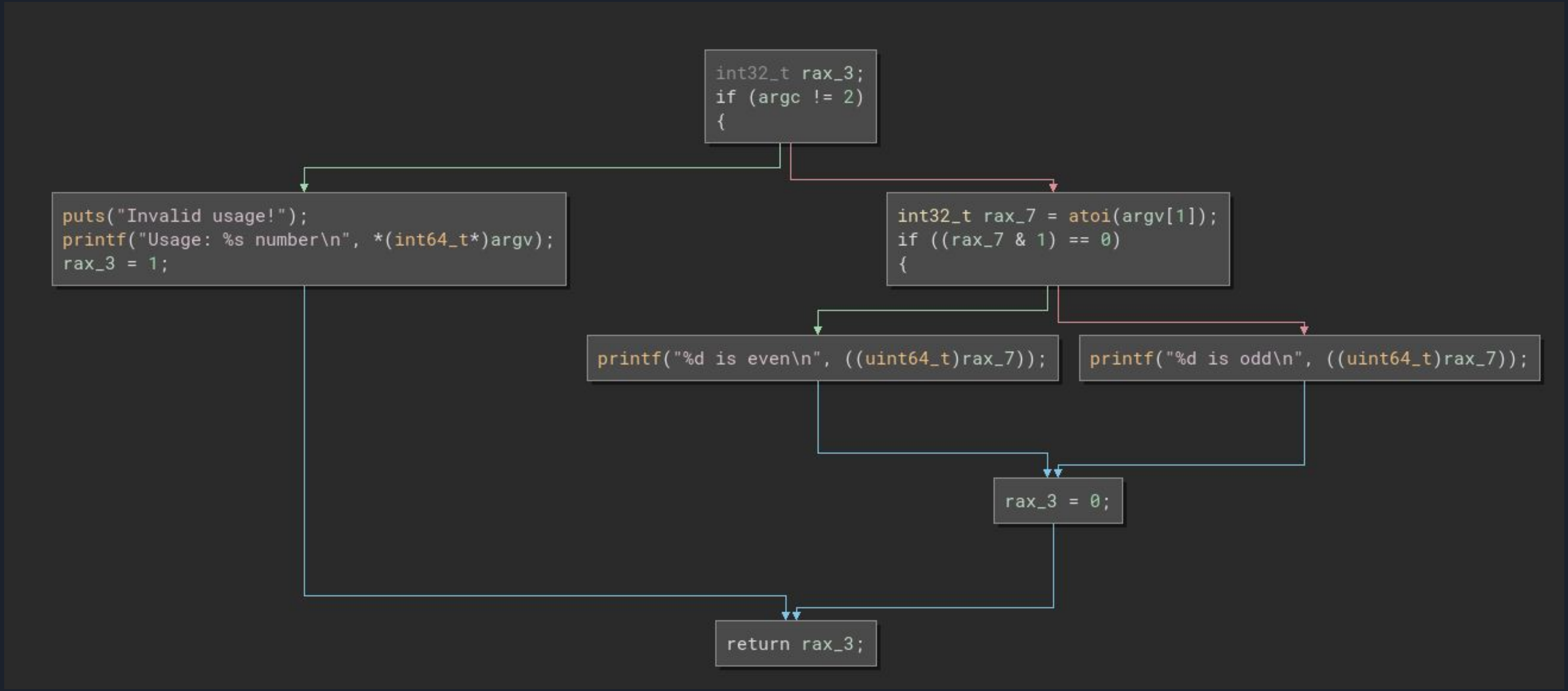
Example



Terminal - mwales@Metroid: ~/checkouts/WildcatCSClub/slides/level5
File Edit View Terminal Tabs Help

```
mwales@Metroid:~/checkouts/WildcatCSClub/slides/level5$ cat odd.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
        if (argc != 2)
        {
                printf("Invalid usage!\n");
                printf("Usage: %s number\n", argv[0]);
                return 1;
        }

        int userNumber = atoi(argv[1]);

        if (userNumber % 2)
                printf("%d is odd\n", userNumber);
        else
                printf("%d is even\n", userNumber);

        return 0;
}
mwales@Metroid:~/checkouts/WildcatCSClub/slides/level5$ 
```

Binary Ninja Graph View



```
int32_t rax_3;
if (argc != 2)
{
```

```
puts("Invalid usage!");
printf("Usage: %s number\n", *(int64_t*)argv);
rax_3 = 1;
```

```
int32_t rax_7 = atoi(argv[1]);
if ((rax_7 & 1) == 0)
{
```

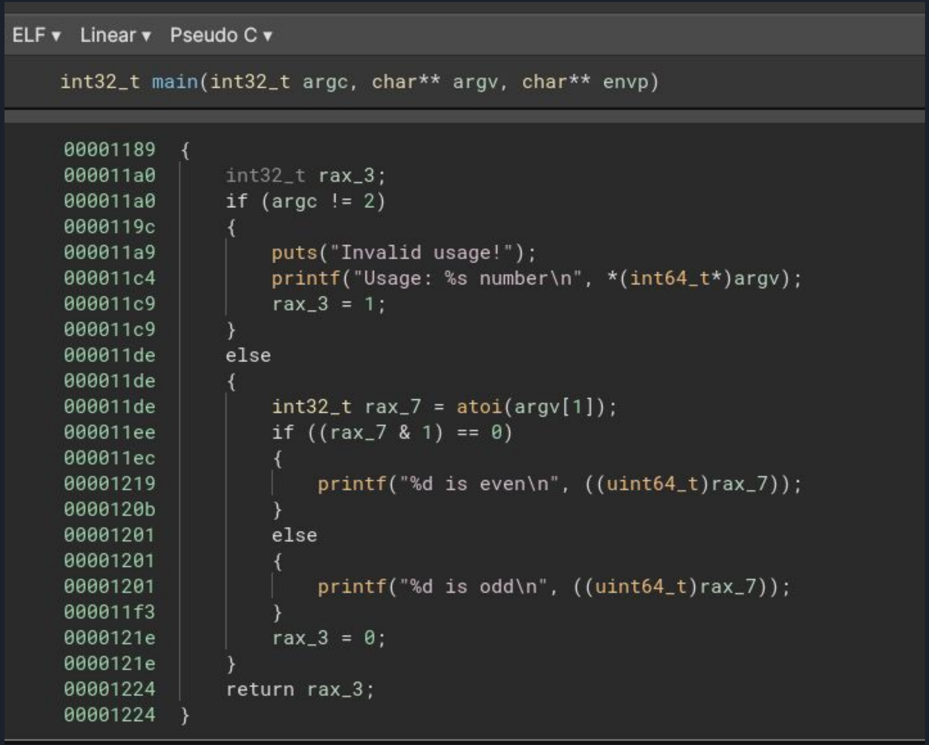
```
printf("%d is even\n", ((uint64_t)rax_7));
```

```
printf("%d is odd\n", ((uint64_t)rax_7));
```

```
rax_3 = 0;
```

```
return rax_3;
```

Binary Ninja Linear View



```
int32_t main(int32_t argc, char** argv, char** envp)

00001189  {
000011a0      int32_t rax_3;
000011a0      if (argc != 2)
0000119c      {
000011a9          puts("Invalid usage!");
000011c4          printf("Usage: %s number\n", *(int64_t*)argv);
000011c9          rax_3 = 1;
000011c9      }
000011de      else
000011de      {
000011de          int32_t rax_7 = atoi(argv[1]);
000011ee          if ((rax_7 & 1) == 0)
000011ec          {
00001219              printf("%d is even\n", ((uint64_t)rax_7));
0000120b          }
00001201          else
00001201          {
00001201              printf("%d is odd\n", ((uint64_t)rax_7));
000011f3          }
0000121e          rax_3 = 0;
0000121e      }
00001224      return rax_3;
00001224  }
```

Links

- https://cybercodequest.lockheedmartin.com/html/dspRegistration.cfm
- https://chat.openai.com/chat