NOVA CTF {2023} - NOC List
[REVERSE ENGINEERING CHALLENGE]

Given Description:

Good morning, Mr. Jim Phelps. The man you're looking at is Alexander Golitsyn, an attaché at our embassy in Prague. He is also a traitor. He has stolen one half of the CIA NOC list, a record of all our deep-cover agents working in Eastern Europe. For security reasons, the NOC list is divided in two. The portion that Golitsyn already has contains code names, but this half is useless without its mate, which matches the code names with their true names. It is this half which Golitsyn plans to steal from the embassy during a reception tomorrow night. Your mission, Jim, should you choose to accept it, is to obtain photographic proof of the theft, shadow Golitsyn to his buyer, and apprehend them both. We've already dispatched a team selected from your usual group. Sarah Davies is already undercover, Jack Harmon can hack into any security system, Hannah Williams will handle surveillance, your wife Claire will cover transport, and Ethan Hunt will be your pointman as usual. He is now in Kiev and will rendezvous in Prague at a safehouse of your choosing. As always, should you or any member of your I.M. force be caught or killed, the Secretary will disavow all knowledge of your actions. This tape will self-destruct in five seconds. Good luck, Jim.

Solution:

The given code for the challenge is:

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <stdlib.h>

char * enc(const char * key)
{
    char * result = (char *) malloc(64);

    int length = strlen(key);
    unsigned char seed = 0x48;
    for(int i = 0; i < length; i++)
    {
        result[i] = 48+(seed*key[i] + seed*12 + 17)%70;
        seed = result[i];
    }
    return result;
}

_Bool verify_key(char * key)
{
    if(strlen(key) < 10 || strlen(key) > 64)
```

```
    {
        printf("Key Invalid!");
      return false;
    }
    char * result = enc(key);
    char * compare = "afW@I?rYoFA";
    return !strcmp(compare, result);
}

int main()
{
    setvbuf(stdout,0, _IONBF,0);
    printf("\nEthan Hunt found the NOC list part 1 key: [wEdERc_u<jj-F]  \n");
    printf("\nCan you help him find the key for NOC list part 2. Enter the key: \n");
    char pkey[65];
    fgets((char*)pkey, 65, stdin);
    if (verify_key(pkey))
    {
        FILE* infile = fopen("flag.txt", "r");
        if (infile == NULL)
        {
            printf("its Too bad the flag is only on the remote server!\n");
            return 0;
        }
        char output[100];
        fgets(output, 100, infile);
        printf("%s", output);
    }
}
```

So the decoding function for the enc function is:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char * dec(const char * key);

int main()
{
    char * encoded = "[OIonU2_<__nK<KsK";   //Change this
    char * decoded = dec(encoded);
    printf("Encoded string: %s\n", encoded);
    printf("Decoded string: %s\n", decoded);
    free(decoded);
```

```
    return 0;
}

char * dec(const char * key)
{
    char * result = (char *) malloc(64);
    int length = strlen(key);
    unsigned char seed = 0x48;
    for(int i = 0; i < length; i++)
    {
        int value = key[i] - 48;
        result[i] = (value - seed*12 - 17) / seed;
        seed = value;
    }
    return result;
}
```

To solve it in it’s way:
Key is: afW@I?rYoFA - J1m_Ph31p5

Solving it - the flag is NOVA{1_n3v32_13f7_7h3_1mf}