还行的一道题

```
IDA View-A | Pseudocode-B | Pseudocode-A | Findcrypt results | Hex View-1 | Structure
    a2 = (char **)((char *)v6 + 4 * i);
    __isoc99_scanf("%d", a2);
  }
  v7[0] = 0LL;
  v7[1] = 0LL;
  v7[2] = 0LL;
  v7[3] = 0LL;
  v7[4] = 0LL;
  for ( j = 0; j <= 2; ++j )
  {
    dword_601078 = v6[j];
    dword_60107C = HIDWORD(v6[j]);
    a2 = (char **)dword_601060;
    sub_400686((unsigned int *)&dword_601078, dword_601060);
    LODWORD(v7[j]) = dword_601078;
    HIDWORD(v7[j]) = dword_60107C;
  }
  if ( (unsigned int)sub_400770(v7, a2) != 1 )
  {
    puts("NO NO NO~ ");
    exit(0);
  }
  puts("Congratulation!\n");
  puts("You seccess half\n");
  puts("Do not forget to change input to hex and combine~\n");
  puts("ByeBye");
  return 0LL;
}
```

没有啥混淆，check在sub_400770里面

```
__int64 __fastcall sub_400770(_DWORD *a1)
{
  __int64 result; // rax

  if ( a1[2] - a1[3] == 2225223423LL
    && a1[3] + a1[4] == 4201428739LL
    && a1[2] - a1[4] == 1121399208LL
    && *a1 == -548868226
    && a1[5] == -2064448480
    && a1[1] == 550153460 )
  {
    puts("good!");
    result = 1LL;
  }
  else
  {
    puts("Wrong!");
    result = 0LL;
  }
  return result;
}
```

其实是求个方程组，可以得到6个输入经过变换后的值。

```
  v7[4] = 0LL;
  for ( j = 0; j <= 2; ++j )
  {
    dword_601078 = v6[j];
    dword_60107C = HIDWORD(v6[j]);
    a2 = (char **)dword_601060;
    sub_400686((unsigned int *)&dword_601078, dword_601060);
    LODWORD(v7[j]) = dword_601078;
    HIDWORD(v7[j]) = dword_60107C;
  }
```

重点对于输入的每2个数字进行了变换。

变换在sub_400686里面

```
__int64 __fastcall sub_400686(unsigned int *a1, _DWORD *a2)
{
  __int64 result; // rax
  unsigned int v3; // [rsp+1Ch] [rbp-24h]
  unsigned int v4; // [rsp+20h] [rbp-20h]
  int v5; // [rsp+24h] [rbp-1Ch]
  unsigned int i; // [rsp+28h] [rbp-18h]

  v3 = *a1;
  v4 = a1[1];
  v5 = 0;
  for ( i = 0; i <= 0x3F; ++i )
  {
    v5 += 1166789954;
    v3 += (v4 + v5 + 11) ^ ((v4 << 6) + *a2) ^ ((v4 >> 9) + a2[1]) ^ 0x20;
    v4 += (v3 + v5 + 20) ^ ((v3 << 6) + a2[2]) ^ ((v3 >> 9) + a2[3]) ^ 0x10;
  }
  *a1 = v3;
  result = v4;
  a1[1] = v4;
  return result;
}
```

可以看到那个操作
一开始我以为不能解
后来看了其他人的wp后发现确实可以逆着来

```
//#include <stdio.h>
//#include <stdlib.h>
int main(){
unsigned int check[6];
check[0]=-548868226;
check[5]=-2064448480;
check[1]=550153460;
check[2]=(2225223423+4201428739+1121399208)/2;
check[3]=1121399208+4201428739-check[2];
check[4]= 2225223423+4201428739-check[2];
unsigned int v3 = 1;
unsigned int v4 = 2;
int i;
int j;
for(i=0;i<6;i++){
printf("%u\n",check[i]);
}
for(j=0;j<3;j++){
v3 = check[2j];
```

```
v4 = check[2j+1];
unsigned int v5 = 1166789954;
for(i=0;i<=0x3f;i++){
v4 -= (v3 + v5 + 20) ^ ((v3 << 6) + 3) ^ ((v3 >> 9) + 4) ^ 0x10;
v3 -= (v4 + v5 + 11) ^ ((v4 << 6) + 2) ^ ((v4 >> 9) + 2) ^ 0x20;
v5-= 1166789954;
}
check[2j]=v3;
check[2j+1]=v4;
}
char * flag = (char )check;
for(i=0;i<6;i++){
printf("%c%c%c%c",flag[4i+3],flag[4i+2],flag[4i+1],flag[4i]);
}
system("pause");
return 0;
}
```

在这里可以需要去掉空格