puts("\n[\*] Phase 1\n- Hint: Numbers are always magical!");

v3 = read\_line();

phase1(v3);

int \_\_cdecl phase1(int a1)

{

int result; // eax@3

int v2[6]; // [sp+Ch] [bp-2Ch]@1

int v3; // [sp+24h] [bp-14h]@3

int v4; // [sp+28h] [bp-10h]@1

int i; // [sp+2Ch] [bp-Ch]@5

v4 = \_\_isoc99\_sscanf(a1, "%d %d %d %d %d %d", v2, &v2[1], &v2[2], &v2[3], &v2[4], &v2[5]);

if ( v4 != 6 )

explode\_bomb();

v3 = 1;

result = v2[0];

if ( v2[0] < 1 ) //v2[0] >= 1

explode\_bomb();

for ( i = 1; i <= 5; ++i )

{

result = v2[i];

if ( result != v2[i - 1] + i )

explode\_bomb();

}

return result;

}

// res = v2[1] == v2[i-1] + i = v2[0] + 1 = 2

// res = v2[2] = v2[1] + 2 = 4

// res = v2[3] = v2[2] + 3 = 7

// res = v2[4] = v2[3] + 4 = 11

V2[5] = 11 + 5 = 16

// co the co nhieu dap an

  puts("\n[\*] Phase 2\n- Hint: You must answer your secret question!");

  v4 = read\_line();

  phase2(v4);

"Which city has 3/7 of a chicken and 2/3"

  int \_\_cdecl phase2(int a1)

{

  char \*v1; // ST28\_4@1

  int result; // eax@2

  char \*s1; // [sp+Ch] [bp-1Ch]@1

  char \*s2; // [sp+10h] [bp-18h]@1

  v1 = QUESTIONS[13];

  s2 = ANSWERS[\*(&QA\_MAP + 13)];

  s1 = (char \*)transfer(a1);

  if ( !\*s2 || (result = is\_equal(s1, s2)) == 0 )

    explode\_bomb();

  return result;

}

// CHICAGO

  puts("\n[\*] Phase 3\n- Hint: Many cases make everything so confusing.");

  v5 = read\_line();

  phase3(v5);

  int \_\_cdecl phase3(int a1)

{

  int result; // eax@14

  int v2; // [sp+0h] [bp-18h]@1

  int v3; // [sp+4h] [bp-14h]@1

  int v4; // [sp+8h] [bp-10h]@1

  int v5; // [sp+Ch] [bp-Ch]@1

  v5 = 0;

  v4 = 0;

  v4 = \_\_isoc99\_sscanf(a1, "%d %d", &v3, &v2); // nhap 2 so nguyen

  if ( v4 <= 1 )

    explode\_bomb();

  switch ( v3 ) // v3 = 4

  {

    case 0:

      v5 += 440;

      goto LABEL\_5;

    case 1:

LABEL\_5:

      v5 -= 555;

      goto LABEL\_6;

    case 2:

LABEL\_6:

      v5 += 651;

      goto LABEL\_7;

    case 3:

LABEL\_7:

      v5 -= 501;

      goto LABEL\_8;

    case 4:

LABEL\_8:

      v5 += 501; // v5 = 501

      goto LABEL\_9;

    case 5:

LABEL\_9:

      v5 -= 501; // v5 = 501 – 501 = 0

      goto LABEL\_10;

    case 6:

LABEL\_10:

      v5 += 501; // v5 = 501

      break;

    case 7:

      break;

    default:

      explode\_bomb();

      return result;

  }

// v3 = 4

  v5 -= 501; v5 = -501

  if ( v3 > 5 || (result = v2, v5 != v2) )

    explode\_bomb();

v3 < 5 || res = v2, v5 == v2

res != 501

  return result;

}

// 4 0

 puts("\n[\*] Phase 4\n- Hint: Let's dig in to recursive function :)");

  v6 = read\_line();

  phase4(v6);

  int \_\_cdecl phase4(int a1)

{

  int result; // eax@5

  int v2; // [sp+Ch] [bp-1Ch]@1

  int v3; // [sp+10h] [bp-18h]@1

  int v4; // [sp+14h] [bp-14h]@5

  int v5; // [sp+18h] [bp-10h]@5

  int v6; // [sp+1Ch] [bp-Ch]@1

  v6 = \_\_isoc99\_sscanf(a1, "%d %d", &v2, &v3);

  if ( v6 != 2 || v3 <= 1 || v3 > 4 ) // v3 = 2

    explode\_bomb();

  v5 = 7;

  v4 = func4(7, v3); // v4 = func4(7, 10) = 330

  result = v2;

  if ( v4 != v2 ) // v2 == v4 == 330

    explode\_bomb();

  return result;

}

int \_\_cdecl func4(int a1, int a2)

{

  int result; // eax@2

  int v3; // ebx@5

  if ( a1 > 0 )

  {

    if ( a1 == 1 )

    {

      result = a2;

    }

    else

    {

      v3 = func4(a1 - 1, a2) + a2;

      result = v3 + func4(a1 - 2, a2);

    }

  }

  else

  {

    result = 0;

  }

  return result;

}

Phase 5:

 puts("\n[\*] Phase 5\n-Hint: No hint is also a hint :)");

  s = (char \*)read\_line();

  phase5(s);

  size\_t \_\_cdecl phase5(char \*s)

{

  size\_t result; // eax@1

  int v2; // [sp+8h] [bp-10h]@3

  signed int i; // [sp+Ch] [bp-Ch]@3

  result = strlen(s); // do dai cua chuoi s

  if ( result != 6 ) // s.len = 6

    explode\_bomb();

  v2 = 0;

  for ( i = 0; i <= 5; ++i )

  {

    result = array\_3852[s[i] & 0xF];

    v2 += result;

  }

  if ( v2 != 48 ) // v2 == 48

    explode\_bomb();

  return result;

}

array\_3852 = [2, 10, 6, 1, 12, 16, 9, 3, 4, 7, 14, 5, 11, 8]