제 1회 청소년 화이트해커 경진대회

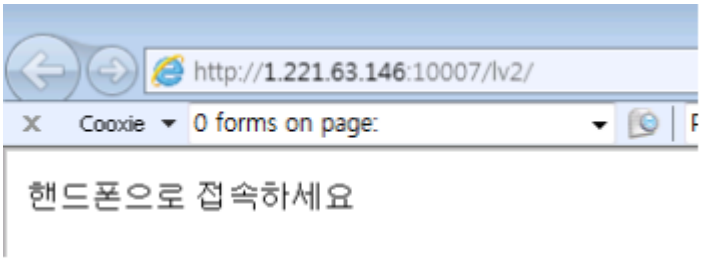
4위 chl5662(최규범)

Level 1

레벨 1에선 계산을 하는 문제였는데, 단순히 계산을 하여 키 값을 인증하였다.
(문제가 닫혀 키 값을 구하지 못함)

Level 2

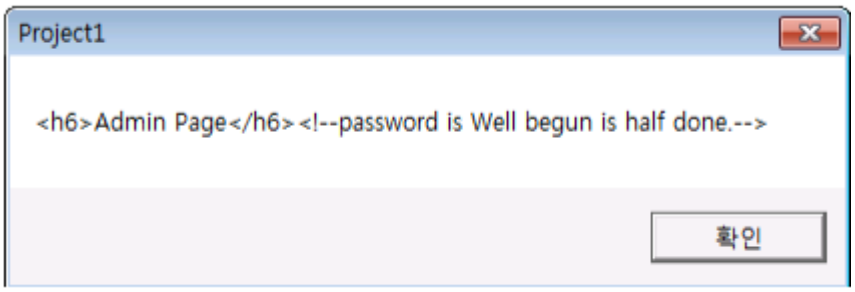
http://1.221.63.146:10007/lv2/
<Level2에서 주어진 주소>



<주어진 주소에 접속한 결과>

해당 주어진 주소를 들어가 보니 다음가 같은 내용이 떴다.
핸드폰으로 접속해 보니 저 내용이 Admin Page로만 바뀌었다.

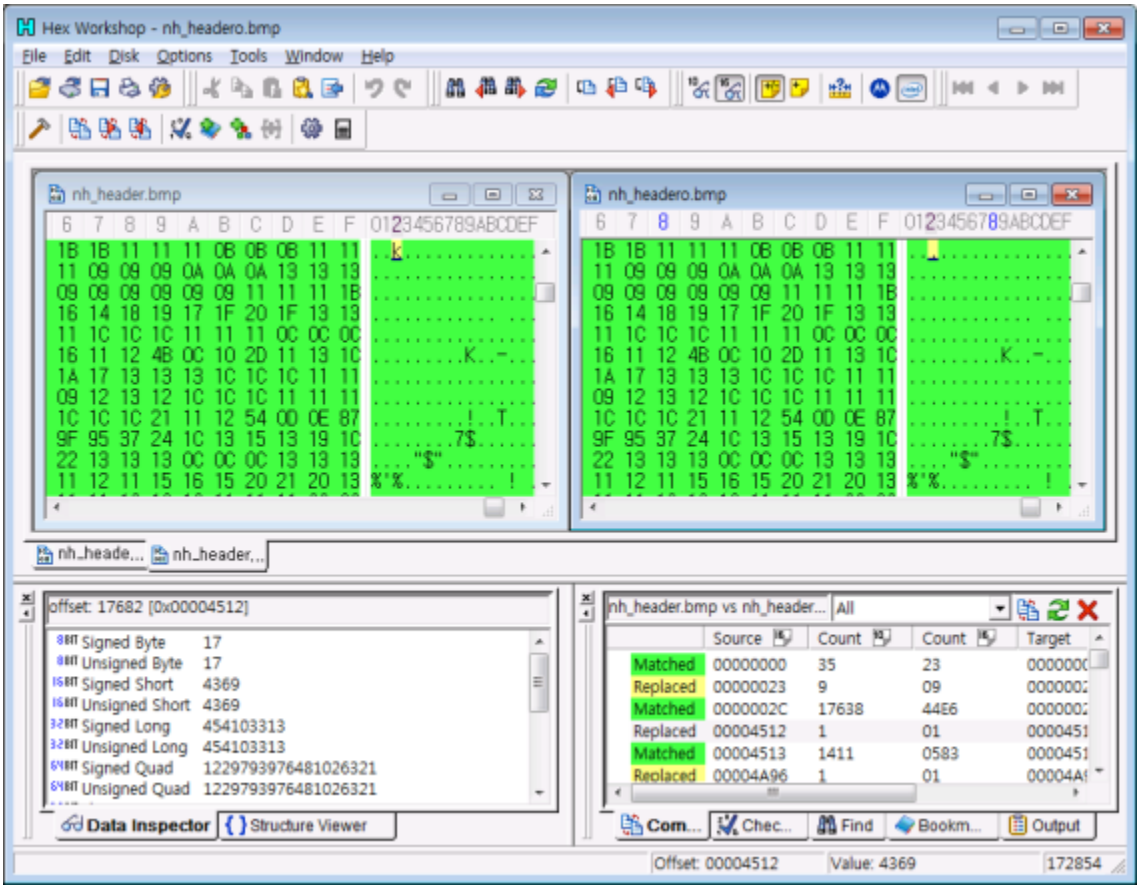
다음과 같은 VB 코드를 작성하여 해당 페이지의 소스를 받아 왔다.



Key : Well begun is half done.Level 3

<레벨3의 배너 사진>

해당 사진을 잘 보면 걸리적 거리는 점 몇 개를 발견 할 수 있었다.
뉴하트 홈페이지 (http://newheart.kr/xe/) 에 접속하여 원본 배너를 구한 뒤,
BMP 파일로 변경 한 후 대조하였다.

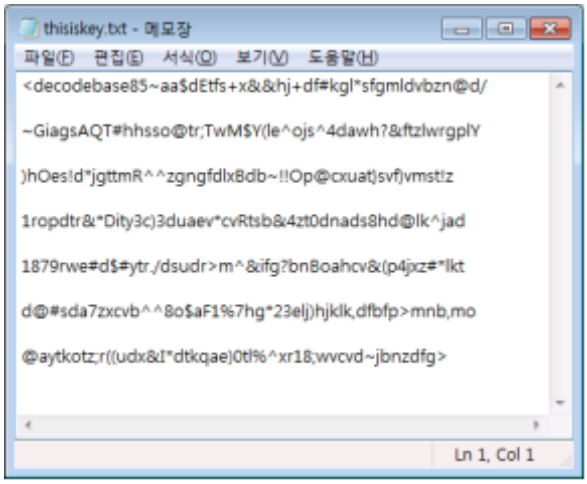


<Hex Workshop의 대조기능을 사용하여 바뀐 부분을 확인>

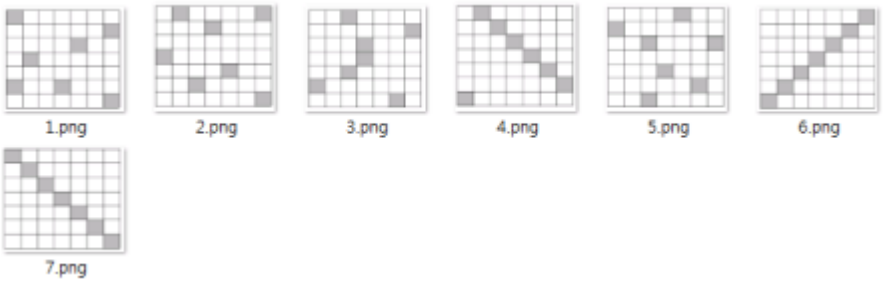
Key : newhe@rt!!

Level 4

레벨 4에서 주어진 압축을 풀어보니, 표 7개와 thisiskey.txt가 있었다.



<thisiskey.txt>



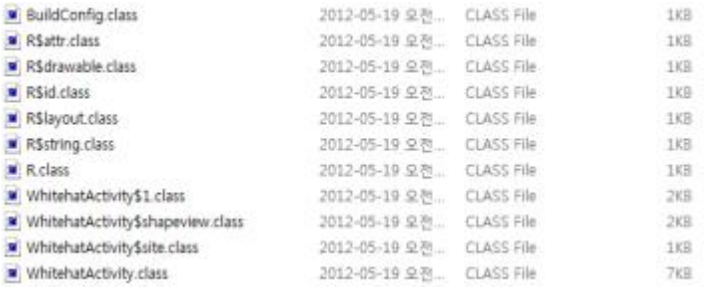
<1~7.png>

표의 크기는 7 * 7 이었고, 마침 thisiskey.txt의 내용도 7줄에 한 줄당 49글자였다.
thisiskey.txt의 내용을 7글자 씩 끊어서 놓은 뒤
각 n.png에서 사진에서 음양이 정해진 부분의 좌표를 (a,b)라고 하면
thisiskey.txt의 n번째 줄 7개로 끊은 것중의 a번째, b번째 글자를 순서대로 배열하였다.
<~E+*g/GAhM4?YORgBOu!rDuR0d@r#drB4#7^F*),>@;I)r~>
<배열한 결과>
결과를 Base85로 디코딩 하면 키 값이 나온다.
(보고서 작성시에 적은 것 이기 때문에 조금 틀린게 있음)

Key : hello_hacking_festival!

Level 5

레벨 5의 apk파일을 dex -> jar로 바꾸어 파일을 보았다.



<jar파일 안의 class파일>

class파일들 중 WhitehatActivity.class를 디컴파일하여 보았더니 decript라는 함수가 있었다.

```
public void df23089ikgdf()
{
   String s = decript("ygbahi?+hih5whhsb1r");
   Toast.makeText(this, s, 1).show();
}

public void oid9059S()
{
   String s   decript("ygbahi?+hih5whhsb1r");
   Toast.makeText(this, s, 1).show();
}

public void oiuhcng()
{
   String s = decript("ygbahi?+hih5whhsb1r");
   Toast.makeText(this, s, 1).show();
}
```

<decript 함수를 호출하는 부분과 파라메터>

```
class Test1
{
   public static void main(String[] args)
   {
      char ai[];
      ai = new char[45];
      char param[] = {'y','g','b','a','h','i','?','+','h','i','h','5','w','h','h','s','b','1','r','W0'};
      ai[0] = 1;
      ai[2] = 34;
          .
          .
          .
      ai[42] = 48;
      ai[43] = 95;
      ai[44] = 38;
      for(int i=0;i<param.length;i++)
      {
         for(int j=0;j<45;j++)
         {
            if("abcdefghiklmnopqrstuvwxyz/.1234567890~_;%+=?".charAt(ai[j])==param[i])
            {
               System.out.printf("%c","abcdefghiklmnopqrstuvwxyz/.1234567890~_;%+=?".charAt(j));
               break;
            }
         }
      }
   }
```

```
C:\Users\WK      \Desktop\ganda>java Test1
diablo3+lol=hellgate
C:\Users\WK      \Desktop\ganda>
```

<decript 함수를 구현 후 실행>

Key : diablo3+lol=hellgate

Level 6

레벨 6에서는 윈도우 모바일 어플리케이션 파일 nhf3.xap가 주어졌다.

<해당 어플리케이션을 실행한 화면>

해당 어플리케이션은 입력한 입력한 번호의 순에 따라 AES 암호화 값을 출력해 준다.

<App xmlns="" Publisher="newheart" Description="134628957" Author="newheart author" Genre="apps.normal" Version="1.0.0.0" RuntimeType="Silverlight" Title="NewHeart" ProductID="{afa6e73c-e753-43c8-9470-b34ce1941e59}">

<WMAppManifest.xml 내용 중 일부>

WMAppManifest.xml를 열어보니 다음과 같은 숫자가 있었고, 해당 번호를 입력한 후 키 값을 얻어내었다.

Key :

u+Vscbgx4hX8Onbrk0dH8Rxcbdg1FnCOH8xn2Uy8aDkoUk4hcHvRK/LGpuMCqQ8N

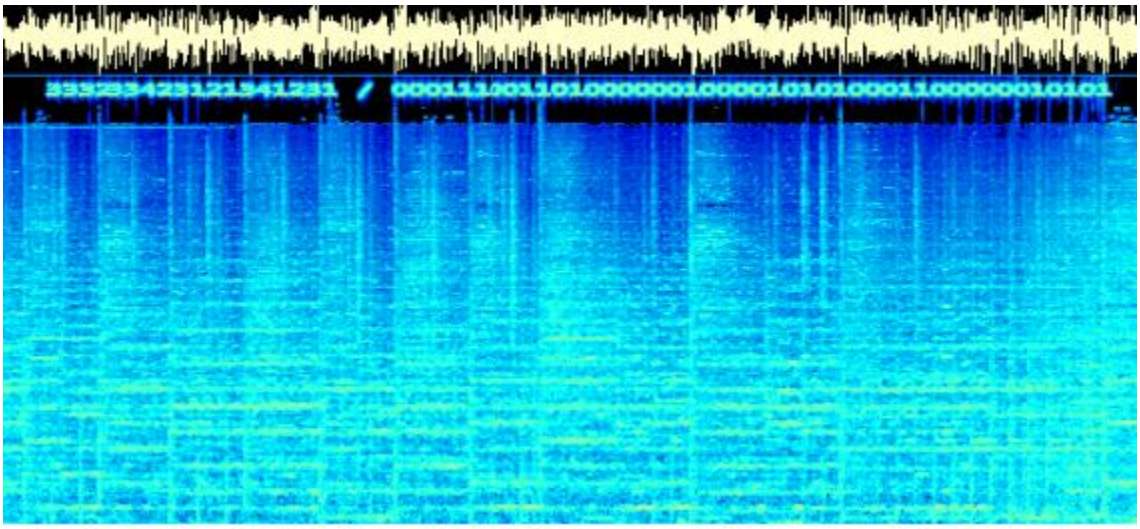
Level 7

레벨 7에서는 룰렛으로 포인트를 얻는 웹페이지가 주어졌다.
포인트를 얻는 페이지의 파라미터에는 타임스탬프와 얻은 포인트 내용이 있었고,
타임스탬프값에 SQL Injection이 가능함을 알게 되었다.
SQL Injection을 통해 경품인 'Answer'을 교환하여 키 값을 얻어내었다.
(문제 서버의 주소를 알아내지 못하여 자세히 쓰지 못함)

Level 8

<Level8의 wavwav.wav>

레벨 8에선 Wave 파일 하나가 주어졌다.



<해당 Wave 파일의 스펙트럼>

Wave 파일의 스펙트럼을 분석해보니 다음과 같은 사진이 글자가 나왔다.

333233423121341231 / 00011100110100000010000101010001100000010101

처음엔 나누기 연산을 하는 줄 알고 헤매었다가, 앞의 숫자들의 합이 뒤의 숫자의 개수와 같은 걸 보고 모스부호로 시도 해 보았다.

... --- ..- -. -.. ... .-.. .. -.- . -. . .-- .... . .- .-. -

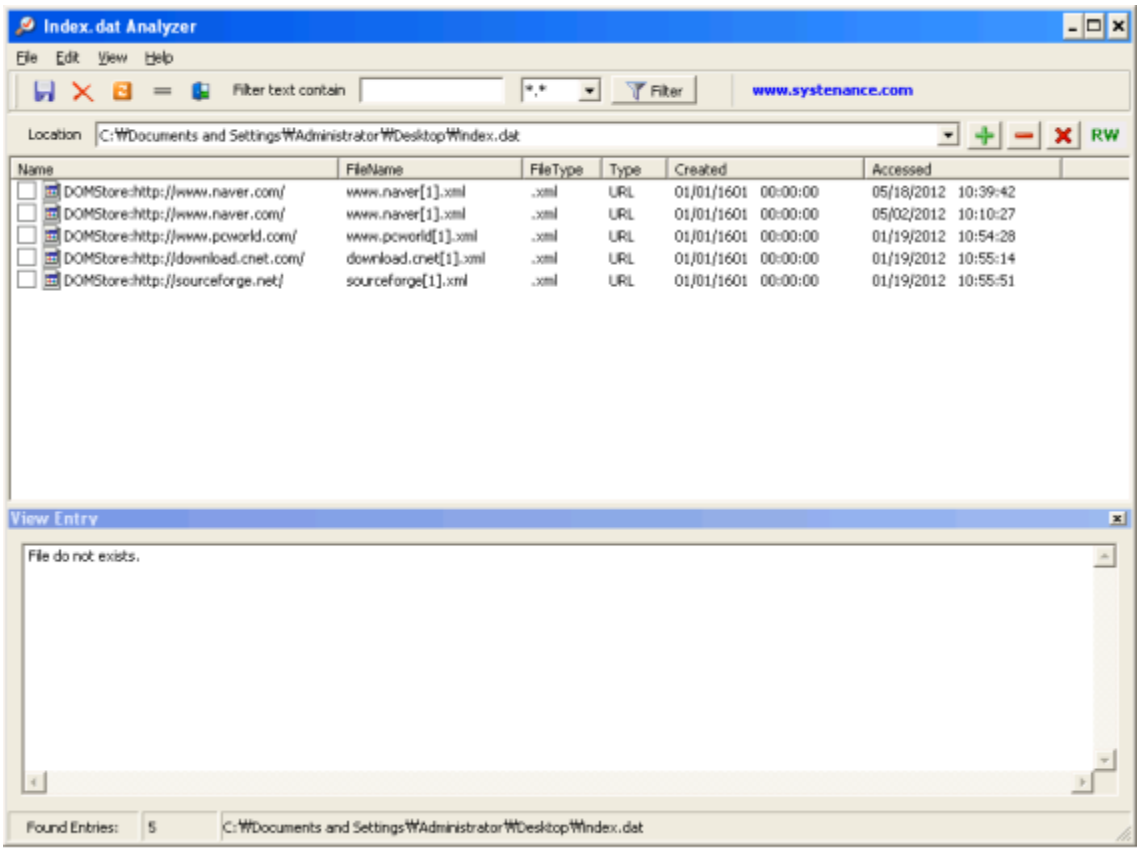
<모스 부호로 변환>

해당 모스 부호를 메시지로 변환하였더니 키 값이 나왔다.

Key : SOUNDSLIKENEWHEART

Level 9

레벨 9에서 주어진 파일을 분석해보니 인터넷 히스토리가 남겨져 있는
index.dat 파일 이었다.



<해당 파일을 분석한 결과>
(Windows 7 x64 에선 작동하지않아, VM Ware에서 캡쳐함)

Index.dat Analyzer 라는 유틸리티 프로그램을 이용하여 키 값을 얻어내었다.

Key : 20120119105514

Level 10

레벨 10에선 안드로이드 이미지파일이 주어지고 범인의 접선 장소와 시간을 찾아야 했다.
data/data/ 폴더로 가서 파일들의 db를 보았으나 원하는 결과를 얻지 못했다.

그러던 중 data/data/com.android.providers.telephony/app_parts 의 파일을 발견했다.

<data/data/com.android.providers.telephony/app_parts 의 파일>

IU_CONCERT_1800_PM_JUNE_02_2012

<PART_13372276170 파일>

파일 헤더는 JPG였기 때문에 확장자를 변경하여 이미지를 보니 키 값이 나왔다.

Key : IU_CONCERT_1800_PM_JUNE_02_2012

Level 11

레벨 11에선 sys파일이 주어졌다.

```
int __stdcall sub_11540(int a1, int a2, int a3, int a4, int a5, int a6)
{
  STRING DestinationString; // [sp+Ch] [bp-164h]@2
  char v8; // [sp+18h] [bp-158h]@3
  LARGE_INTEGER ByteOffset; // [sp+140h] [bp-30h]@3
  HANDLE Handle; // [sp+14Ch] [bp-24h]@3
  OBJECT_ATTRIBUTES ObjectAttributes; // [sp+150h] [bp-20h]@3
  struct _IO_STATUS_BLOCK IoStatusBlock; // [sp+168h] [bp-8h]@3

  if ( a2 == 65664 )
  {
    RtlUnicodeStringToAnsiString(&DestinationString, *(PCUNICODE_STRING *)(a3 + 8), 1u);
    if ( !stricmp(DestinationString.Buffer, "\??\C:\NewHeart\Memo.txt") )
    {
      RtlInitUnicodeString((PUNICODE_STRING)&v8, L"\DosDevices\C:\NewHeart\Memo.txt");
      ObjectAttributes.Length = 24;
      ObjectAttributes.RootDirectory = 0;
      ObjectAttributes.Attributes = 64;
      ObjectAttributes.ObjectName = (PUNICODE_STRING)&v8;
      ObjectAttributes.SecurityDescriptor = 0;
      ObjectAttributes.SecurityQualityOfService = 0;
      ZwCreateFile(&Handle, 0xC0100000u, &ObjectAttributes, &IoStatusBlock, 0, 0x80u, 3u, 3u, 0x60u, 0, 0);
      sub_11180((int)&unk_13008);
      ZwWriteFile(Handle, 0, 0, 0, &IoStatusBlock, byte_130E0, 0x18u, &ByteOffset, 0);
      ZwClose(Handle);
      dword_130C8 = (PKTIMER)ExAllocatePool(0, 0x28u);
      P = ExAllocatePool(0, 0x20u);
      KeInitializeTimer(dword_130C8);
      KeInitializeDpc((PRKDPC)P, (PKDEFERRED_ROUTINE)DeferredRoutine, 0);
      KeSetTimerEx(dword_130C8, (LARGE_INTEGER)-10i64, 2000, (PKDPC)P);
    }
  }
  return dword_130CC(a1, a2, a3, a4, a5, a6);
}
```

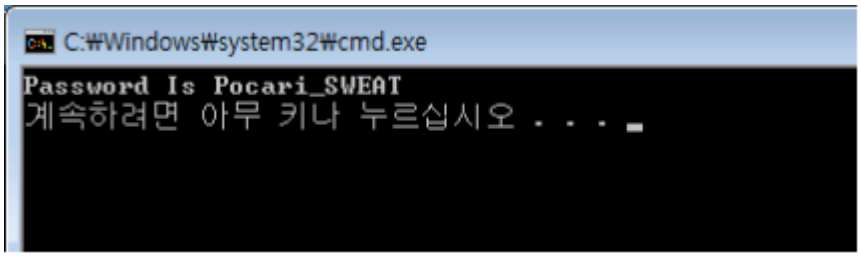
<IDA로 본 P-Code>

IDA로 열어 P-Code를 보았다.

파일 작성을 하는 것 같아, ZWWriteFile의 버퍼를 연산해주는 sub_11180의 P-Code를 가져와 실행가능하도록 수정하였다.

```
char byte_130A0[50] = "";
char byte_130E0[50] = "";
char unk_13008[] = {0xD2,0xD4,0x21,0x48,0x38,0xFD,0x15,0xAA,0xBE,0x3E,0xD4,0x6D,0x93,0xE7,0xF3,0x87,0x03,0x8E,0xB4,0x48,0x29,
0x92,0xDF,0xAB,0xCD,0xFD,0xFD,0xFD,0xFD,0xDD,0xDD,0xDD,0xF7,0x37,0x7F,0x2F,0x1C,0xFF,0x00,0x00};
void *__stdcall sub_11010(int a1)
{
  void *result; // eax@1
  bool v2; // [sp+0h] [bp-40h]@12
  signed int v3; // [sp+4h] [bp-3Ch]@7
  char v4[50]; // [sp+8h] [bp-38h]@1
  int i; // [sp+34h] [bp-Ch]@1
  double v8; // [sp+38h] [bp-8h]@1
  int v9; // [sp+40h] [bp+0h]@1

  v8 = 0.200013;
  for ( i = 0; i < 10000; ++i )
    v8 = 4.0 * v8 * (1.0 - v8);
  for ( i = 0; i < 24; ++i )
  {
    result = (void *)i;
    v4[i] = 0;
  }
  v3 = -1;
  for ( i = 0; i < 8 * a1; ++i )
  {
    if ( !(i % 8) )
      ++v3;
    v4[v3] += 2;
    v2 = v8 > 0.5;
    result = (void *)(v2 | v4[v3]);
    v4[v3] = (char)result;
    v8 = 4.0 * v8 * (1.0 - v8);
  }
  for ( i = 0; i < 40; ++i )
  {
    result = (void *)i;
    byte_130A0[i] = v4[i];
  }
  return result;
}

char *__stdcall sub_11180(int a1)
{
  char v1; // ST13_1@2
  int v3; // [sp+8h] [bp-40h]@1
  int v4; // [sp+Ch] [bp-3Ch]@1
  char v5[50]; // [sp+18h] [bp-30h]@3
  int i; // [sp+44h] [bp-4h]@3
  int v9; // [sp+48h] [bp+0h]@1

  v4 = a1;
  v3 = a1 + 1;
  do
    v1 = *(BYTE *)v4++;
  while ( v1 );
  sub_11010(v4 - v3);
  for ( i = 0; i < v4 - v3; ++i )
    v5[i] = byte_130A0[i] ^ *(BYTE *)(i + a1);
  for ( i = 0; i < 40; ++i )
    byte_130E0[i] = v5[i];
  printf("%s\n",v5);
  return v5;
}

int _tmain(int argc, _TCHAR* argv[])
{
    sub_11180((int)&unk_13008);
    return 0;
}
```

<소스코드>



<실행결과>

Key：Pocari_SWEAT