```
/* This file was generated by the Hex-Rays decompiler version 7.7.0.220118.
   Copyright (c) 2007-2021 Hex-Rays <info@hex-rays.com>

   Detected compiler: Visual C++
*/

#include <windows.h>
#include <defs.h>


//-------------------------------------------------------------------------
// Function declarations

#define __thiscall __cdecl // Test compile in C mode

_DWORD *__thiscall sub_E3139A(_DWORD *this, int, int);
_DWORD *__thiscall sub_E313C7(_DWORD *this, int);
int __thiscall sub_E31401(_DWORD *this);
_DWORD *__thiscall sub_E31449(_DWORD *this, char);
_DWORD *__thiscall sub_E31473(_DWORD *this, int);
_DWORD *__thiscall sub_E31497(_DWORD *this, char);
int __thiscall sub_E314C1(_DWORD *this);
_DWORD *__thiscall sub_E314D2(_DWORD *this);
_DWORD *__thiscall sub_E314F4(_DWORD *this, char);
int __thiscall sub_E3151E(_DWORD *this);
_DWORD *__thiscall sub_E3154D(_DWORD *this, int);
_DWORD *__thiscall sub_E3156F(_DWORD *this, int);
// void *__cdecl operator new(size_t Size, int, const char *, int); idb
_DWORD *__thiscall sub_E31674(_DWORD *this);
_DWORD *__thiscall sub_E316B6(_DWORD *this, int);
int __thiscall sub_E31721(_DWORD *this, unsigned int);
void *__thiscall jCrucialEncode(void *this);
// int *__userpurge sub_E318BC@<eax>(int *@<ecx>, int@<ebp>, _DWORD *);
int __thiscall sub_E31984(_DWORD *this, int);
int *__thiscall sub_E319DE(int *this);
int __thiscall eCrucialEncode(unsigned __int16 *this);
__int16 *__thiscall aCrucialEncode(__int16 *this, __int16);
int *__thiscall sub_E31DDF(int *this, int);
int __thiscall sub_E31E05(_DWORD *this, int);
void __thiscall sub_E31E40(int *this, unsigned int);
int *__thiscall sub_E31E5B(int *this);
_DWORD *__thiscall sub_E31F52(_DWORD *this);
int __thiscall sub_E31FBF(_DWORD *this);
char *__thiscall sub_E3208B(_DWORD *this, int);
void __thiscall gEncode(void **this);
void hEncode();
__int16 *__thiscall sub_E321D4(unsigned __int16 *this, __int16 *, unsigned __int16 *);
void *__thiscall kEncode(void *this);
_BYTE *__thiscall mEncode(void **this);
_DWORD *__thiscall sub_E324C0(_DWORD *this);
void __thiscall sub_E325B7(void *this, int, int);
int __thiscall rEncode(void *this, int a2, int a3, int a4);
_DWORD *__thiscall sub_E32C45(_DWORD *this, int);
void sub_E32C63();
int __cdecl sub_E32D70(_DWORD); // weak
```

```
int __thiscall sub_E32E6F(void *this, int, int, int);
_DWORD *__thiscall sub_E32EC3(_DWORD *this);
int __cdecl tEncode(int, int, int, int);
unsigned int __cdecl sub_E33533(unsigned int);
int __cdecl sub_E335AC(int, int, int);
void __cdecl sub_E35312(void *Block);
int __cdecl UserMathErrorFunction();
// unsigned __int64 __usercall sub_E363A3@<edx:eax>(unsigned __int64@<edx:eax>);
// int std::locale::_Getgloballocale(void); weak

//----------------------------------------------------------------------
// Data declarations

_UNKNOWN loc_E364CA; // weak
void *std::bad_alloc::‘vftable’ = &sub_E31497; // weak
void *std::bad_array_new_length::‘vftable’ = &sub_E314F4; // weak
void *std::bad_cast::‘vftable’ = &sub_E31497; // weak


//----- (00E31000) --------------------------------------------------------
int sub_401000()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "https://discord.gg/fmhw85T5zM");
  v2 = 0;
  sub_E318BC(dword_E3944C, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E36612);
}
// E3944C: using guessed type int dword_40944C[3];

//----- (00E3105D) --------------------------------------------------------
int sub_40105D()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "V1rtu4lAll0c");
  v2 = 0;
  sub_E318BC(dword_E39458, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E36621);
}
// E39458: using guessed type int dword_409458[3];

//----- (00E310BA) --------------------------------------------------------
int sub_4010BA()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
```

```
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "jpofwejfdslfkjdslkfghiphap332oiu");
  v2 = 0;
  sub_E318BC(dword_E394A0, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E36630);
}
// E394A0: using guessed type int dword_4094A0[3];

//----- (00E31117) --------------------------------------------------------
int sub_401117()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "Ohu3mNdslwoxfedlo34");
  v2 = 0;
  sub_E318BC(dword_E39440, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E3663F);
}
// E39440: using guessed type int dword_409440[3];

//----- (00E31174) --------------------------------------------------------
int sub_401174()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "2-mk43xxy0.1k");
  v2 = 0;
  sub_E318BC(dword_E39464, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E3664E);
}
// E39464: using guessed type int dword_409464[3];

//----- (00E311D1) --------------------------------------------------------
int sub_4011D1()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, ";32;42;65;33;91;52");
  v2 = 0;
  sub_E318BC(dword_E39494, (int)&savedregs, v1);
  v2 = -1;
```

```
  gEncode(v1);
  return atexit(sub_E3665D);
}
// E39494: using guessed type int dword_409494[3];

//----- (00E3122E) --------------------------------------------------------
int sub_40122E()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "S‘=d‘Y9{]D}_vK$#.");
  v2 = 0;
  sub_E318BC(dword_E39434, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E3666C);
}
// E39434: using guessed type int dword_409434[3];

//----- (00E3128B) --------------------------------------------------------
int sub_40128B()
{
  void *v1[6]; // [esp+0h] [ebp-24h] BYREF
  int v2; // [esp+20h] [ebp-4h]
  int savedregs; // [esp+24h] [ebp+0h] BYREF

  sub_E320BA((unsigned int *)v1, "xxxxxGot7.5_HUH?98rjoi2r3oifjdsoigfogdfs");
  v2 = 0;
  sub_E318BC(dword_E39488, (int)&savedregs, v1);
  v2 = -1;
  gEncode(v1);
  return atexit(sub_E3667B);
}
// E39488: using guessed type int dword_409488[3];

//----- (00E312E8) --------------------------------------------------------
int sub_4012E8()
{
  sub_E320BA(dword_E39470, "H4ndy51mpL30bFusC4tI0NL1bR4RybYM3mB3R_TH4nKs");
  return atexit(sub_E3668A);
}
// E39470: using guessed type unsigned int dword_409470[6];

//----- (00E31307) --------------------------------------------------------
int sub_401307()
{
  return atexit(sub_E36699);
}

//----- (00E31328) --------------------------------------------------------
char sub_401328()
{
  return 0;
```

```
}

//----- (00E3132F) --------------------------------------------------------
void *sub_40132F()
{
  return &unk_E39428;
}

//----- (00E31339) --------------------------------------------------------
int __cdecl sub_401339(FILE *Stream, char *Format, _locale_t Locale, va_list ArgList)
{
  unsigned __int64 *v4; // eax

  v4 = (unsigned __int64 *)sub_E3132F();
  return _stdio_common_vfprintf(*v4, Stream, Format, Locale, ArgList);
}

//----- (00E3135D) --------------------------------------------------------
int sub_40135D(char *a1, ...)
{
  FILE *Stream; // [esp+4h] [ebp-Ch]
  va_list va; // [esp+1Ch] [ebp+Ch] BYREF

  va_start(va, a1);
  Stream = _acrt_iob_func(1u);
  return sub_E31339(Stream, a1, 0, va);
}

//----- (00E3139A) --------------------------------------------------------
_DWORD *__thiscall sub_40139A(_DWORD *this, int a2, int a3)
{
  _DWORD *v3; // ecx

  *this = &std::exception::`vftable';
  v3 = this + 1;
  *v3 = 0;
  v3[1] = 0;
  this[1] = a2;
  return this;
}
// E371BC: using guessed type void *std::exception::`vftable';

//----- (00E313C7) --------------------------------------------------------
_DWORD *__thiscall sub_4013C7(_DWORD *this, int a2)
{
  _DWORD *v2; // ecx

  *this = &std::exception::`vftable';
  v2 = this + 1;
  *v2 = 0;
  v2[1] = 0;
  _std_exception_copy(a2 + 4, v2);
  return this;
}
// E370AC: using guessed type int __cdecl _std_exception_copy(_DWORD, _DWORD);
```

```
// E371BC: using guessed type void *std::exception::‘vftable’;

//----- (00E31401) --------------------------------------------------------
int __thiscall sub_401401(_DWORD *this)
{
  *this = &std::exception::‘vftable’;
  return _std_exception_destroy(this + 1);
}
// E370B8: using guessed type int __cdecl _std_exception_destroy(_DWORD);
// E371BC: using guessed type void *std::exception::‘vftable’;

//----- (00E31421) --------------------------------------------------------
const char *__thiscall sub_401421(_DWORD *this)
{
  if ( this[1] )
    return (const char *)this[1];
  else
    return "Unknown exception";
}

//----- (00E31449) --------------------------------------------------------
_DWORD *__thiscall sub_E31449(_DWORD *this, char a2)
{
  sub_E31401(this);
  if ( (a2 & 1) != 0 )
    sub_E35312(this);
  return this;
}

//----- (00E31473) --------------------------------------------------------
_DWORD *__thiscall sub_E31473(_DWORD *this, int a2)
{
  sub_E3139A(this, a2, 1);
  *this = &std::bad_alloc::‘vftable’;
  return this;
}
// E371C8: using guessed type void *std::bad_alloc::‘vftable’;

//----- (00E31497) --------------------------------------------------------
_DWORD *__thiscall sub_E31497(_DWORD *this, char a2)
{
  sub_E314C1(this);
  if ( (a2 & 1) != 0 )
    sub_E35312(this);
  return this;
}

//----- (00E314C1) --------------------------------------------------------
int __thiscall sub_E314C1(_DWORD *this)
{
  return sub_E31401(this);
}

//----- (00E314D2) --------------------------------------------------------
_DWORD *__thiscall sub_E314D2(_DWORD *this)
```

```
{
  sub_E31473(this, (int)"bad array new length");
  *this = &std::bad_array_new_length::‘vftable’;
  return this;
}
// E371E4: using guessed type void *std::bad_array_new_length::‘vftable’;

//----- (00E314F4) --------------------------------------------------------
_DWORD *__thiscall sub_E314F4(_DWORD *this, char a2)
{
  sub_E3151E(this);
  if ( (a2 & 1) != 0 )
    sub_E35312(this);
  return this;
}

//----- (00E3151E) --------------------------------------------------------
int __thiscall sub_E3151E(_DWORD *this)
{
  return sub_E314C1(this);
}

//----- (00E3152F) --------------------------------------------------------
void __cdecl __noreturn sub_40152F()
{
  int pExceptionObject[3]; // [esp+0h] [ebp-Ch] BYREF

  sub_E314D2(pExceptionObject);
  CxxThrowException(pExceptionObject, (_ThrowInfo *)&_TI3_AVbad_array_new_length_std__);
}
// E3152F: using guessed type _DWORD pExceptionObject[3];

//----- (00E3154D) --------------------------------------------------------
_DWORD *__thiscall sub_E3154D(_DWORD *this, int a2)
{
  sub_E3156F(this, a2);
  *this = &std::bad_array_new_length::‘vftable’;
  return this;
}
// E371E4: using guessed type void *std::bad_array_new_length::‘vftable’;

//----- (00E3156F) --------------------------------------------------------
_DWORD *__thiscall sub_E3156F(_DWORD *this, int a2)
{
  sub_E313C7(this, a2);
  *this = &std::bad_alloc::‘vftable’;
  return this;
}
// E371C8: using guessed type void *std::bad_alloc::‘vftable’;

//----- (00E315B6) --------------------------------------------------------
__int64 sub_4015B6()
{
  return 0i64;
}
```

```
//----- (00E315BF) --------------------------------------------------------
__int64 sub_4015BF()
{
  return -1i64;
}

//----- (00E315D8) --------------------------------------------------------
_DWORD *__cdecl sub_4015D8(_DWORD *a1, _DWORD *a2)
{
  _DWORD *result; // eax
  int v3; // [esp+8h] [ebp-8h]
  unsigned int v4; // [esp+Ch] [ebp-4h]

  *a2 += 35;
  v3 = *(_DWORD *)(*a1 - 4);
  v4 = *a1 - v3;
  if ( v4 < 4 || v4 > 0x23 )
    invalid_parameter_noinfo_noreturn();
  result = a1;
  *a1 = v3;
  return result;
}

//----- (00E31642) --------------------------------------------------------
void aEncode()
{
  ;
}

//----- (00E3164B) --------------------------------------------------------
void __stdcall sub_40164B(int a1)
{
  ;
}

//----- (00E31656) --------------------------------------------------------
void *__thiscall bEncode(void *this, int a2, int a3)
{
  return this;
}

//----- (00E31664) --------------------------------------------------------
void __noreturn sub_401664()
{
  std::_Xlength_error("string too long");
}
// E3706C: using guessed type void __cdecl __noreturn std::_Xlength_error(const char *);

//----- (00E31674) --------------------------------------------------------
_DWORD *__thiscall sub_E31674(_DWORD *this)
{
  sub_E3139A(this, (int)"bad cast", 1);
  *this = &std::bad_cast::‘vftable’;
  return this;
```

```
}
// E373D4: using guessed type void *std::bad_cast::‘vftable’;

//----- (00E31698) --------------------------------------------------------
void __noreturn sub_401698()
{
  int pExceptionObject[3]; // [esp+0h] [ebp-Ch] BYREF

  sub_E31674(pExceptionObject);
  CxxThrowException(pExceptionObject, (_ThrowInfo *)&_TI2_AVbad_cast_std__);
}
// E31698: using guessed type _DWORD pExceptionObject[3];

//----- (00E316B6) --------------------------------------------------------
_DWORD *__thiscall sub_E316B6(_DWORD *this, int a2)
{
  sub_E313C7(this, a2);
  *this = &std::bad_cast::‘vftable’;
  return this;
}
// E373D4: using guessed type void *std::bad_cast::‘vftable’;

//----- (00E316D8) --------------------------------------------------------
int (__thiscall ***__thiscall sub_4016D8(_DWORD **this))(_DWORD, int)
{
  int (__thiscall ***result)(_DWORD, int); // eax

  result = (int (__thiscall ***)(_DWORD, int))this;
  if ( this[1] )
  {
    result = (int (__thiscall ***)(_DWORD, int))(*(int (__thiscall **)(_DWORD *))(*this[1] + 8))(this[1]);
    if ( result )
      return (int (__thiscall ***)(_DWORD, int))(**result)(result, 1);
  }
  return result;
}

//----- (00E31721) --------------------------------------------------------
int __thiscall sub_E31721(_DWORD *this, unsigned int a2)
{
  int v3; // [esp+0h] [ebp-10h]
  int v4; // [esp+8h] [ebp-8h]

  if ( a2 >= *(_DWORD *)(this[1] + 12) )
    v4 = 0;
  else
    v4 = *(_DWORD *)(*(_DWORD *)(this[1] + 8) + 4 * a2);
  if ( v4 || !*(_BYTE *)(this[1] + 20) )
    return v4;
  v3 = std::locale::_Getgloballocale();
  if ( a2 >= *(_DWORD *)(v3 + 12) )
    return 0;
  else
    return *(_DWORD *)(*(_DWORD *)(v3 + 8) + 4 * a2);
}
```

```
// E37080: using guessed type int std::locale::_Getgloballocale(void);

//----- (00E31797) --------------------------------------------------------
_DWORD *__cdecl sub_401797(_DWORD *a1)
{
  _DWORD *v1; // eax
  int v3; // [esp+0h] [ebp-10h] BYREF
  __int64 ticks; // [esp+8h] [ebp-8h] BYREF

  ticks = Xtime_get_ticks();
  v1 = sub_E31864(&v3, &ticks);
  sub_E31F83(a1, v1);
  return a1;
}

//----- (00E317C3) --------------------------------------------------------
_DWORD *__cdecl sub_4017C3(_DWORD *a1)
{
  _DWORD *v1; // eax
  int v3; // [esp+0h] [ebp-30h] BYREF
  __int64 v4; // [esp+8h] [ebp-28h] BYREF
  __int64 v5; // [esp+10h] [ebp-20h]
  __int64 v6; // [esp+18h] [ebp-18h]
  __int64 perf_counter; // [esp+20h] [ebp-10h]
  __int64 perf_frequency; // [esp+28h] [ebp-8h]

  perf_frequency = Query_perf_frequency();
  perf_counter = Query_perf_counter();
  v6 = 1000000000 * (perf_counter / perf_frequency);
  v5 = 1000000000 * (perf_counter % perf_frequency) / perf_frequency;
  v4 = v5 + v6;
  v1 = sub_E31864(&v3, &v4);
  sub_E31F83(a1, v1);
  return a1;
}

//----- (00E31864) --------------------------------------------------------
_DWORD *__thiscall sub_401864(_DWORD *this, _DWORD *a2)
{
  int v3; // ecx

  v3 = a2[1];
  *this = *a2;
  this[1] = v3;
  return this;
}

//----- (00E31882) --------------------------------------------------------
__int64 __thiscall sub_401882(void *this)
{
  return *(_QWORD *)this;
}

//----- (00E31893) --------------------------------------------------------
double __thiscall sub_401893(void *this)
```

```
{
  return *(double *)this;
}

//----- (00E318A1) --------------------------------------------------------
void *__thiscall sub_4018A1(void *this)
{
  return this;
}

//----- (00E318AD) --------------------------------------------------------
unsigned int sub_4018AD()
{
  return std::_Random_device();
}
// E37070: using guessed type unsigned int __cdecl std::_Random_device();

//----- (00E318BC) --------------------------------------------------------
int *__userpurge sub_E318BC@<eax>(int *a1@<ecx>, int a2@<ebp>, _DWORD *a3)
{
  int v3; // eax
  unsigned int v4; // eax
  char *v5; // eax
  int *v6; // eax
  int v7; // ecx
  int v8; // eax
  int *v9; // eax
  int v10; // edx
  int v12; // [esp-30h] [ebp-3Ch] BYREF
  int v13; // [esp-28h] [ebp-34h]
  int v14; // [esp-24h] [ebp-30h]
  int *v15; // [esp-20h] [ebp-2Ch]
  int *v16; // [esp-1Ch] [ebp-28h]
  int *v17; // [esp-18h] [ebp-24h]
  unsigned int i; // [esp-14h] [ebp-20h]
  int *v19; // [esp-10h] [ebp-1Ch]
  struct _EXCEPTION_REGISTRATION_RECORD *ExceptionList; // [esp-Ch] [ebp-18h]
  void *v21; // [esp-8h] [ebp-14h]
  int v22; // [esp-4h] [ebp-10h]
  int v23; // [esp+0h] [ebp-Ch]
  int v24; // [esp+4h] [ebp-8h]
  int v25; // [esp+8h] [ebp-4h] BYREF
  int retaddr; // [esp+Ch] [ebp+0h]

  v23 = a2;
  v24 = retaddr;
  v22 = -1;
  v21 = &loc_E364CA;
  ExceptionList = NtCurrentTeb()->NtTib.ExceptionList;
  v19 = &v25;
  v17 = a1;
  sub_E31F52(a1);
  v22 = 0;
  v16 = v17;
  v3 = sub_E31FBF(a3);
```

```
  sub_E31E40(v17, v3);
  for ( i = 0; ; ++i )
  {
    v4 = sub_E31FBF(a3);
    if ( i >= v4 )
      break;
    v5 = sub_E3208B(a3, i);
    v6 = sub_E31DDF(&v12, *v5);
    v7 = *v6;
    v8 = v6[1];
    v13 = v7;
    v14 = v8;
    v15 = v17;
    v9 = (int *)sub_E31E05(v17, i);
    v10 = v14;
    *v9 = v13;
    v9[1] = v10;
  }
  return v17;
}

//----- (00E31984) --------------------------------------------------------
int __thiscall sub_E31984(_DWORD *this, int a2)
{
  return sub_E31E05(this, a2);
}

//----- (00E319A1) --------------------------------------------------------
int *__thiscall cEncode(int *this, int *a2)
{
  dEncode(a2, this);
  return a2;
}

//----- (00E319DE) --------------------------------------------------------
int *__thiscall sub_E319DE(int *this)
{
  return sub_E31E5B(this);
}

//----- (00E319EF) --------------------------------------------------------
int *__cdecl aGetInput(int *a1)
{
  void *UserInput[6]; // [esp+0h] [ebp-28h] BYREF
  int v3; // [esp+18h] [ebp-10h]
  int v4; // [esp+24h] [ebp-4h]
  int savedregs; // [esp+28h] [ebp+0h] BYREF

  v3 = 0;
  printf("Password: ");
  sub_E320BA((unsigned int *)UserInput, "v2");
  v4 = 0;
  bGetInput(std::cin, UserInput);
  sub_E318BC(a1, (int)&savedregs, UserInput);
  v3 |= 1u;
```

```
  v4 = -1;
  gEncode(UserInput);
  return a1;
}

//----- (00E31A6B) ----------------------------------------------------
char __cdecl eEncode(int *a1)
{
  int v1; // esi
  __int16 v3; // ax
  unsigned __int16 v4; // ax
  _DWORD *v5; // eax
  int v6; // esi
  int v7; // edi
  unsigned __int16 v8; // ax
  _DWORD *v9; // eax
  int v10[3]; // [esp+8h] [ebp-54h] BYREF
  int v11[3]; // [esp+14h] [ebp-48h] BYREF
  int v12[3]; // [esp+20h] [ebp-3Ch] BYREF
  __int16 v13; // [esp+2Ch] [ebp-30h] BYREF
  int *v14; // [esp+30h] [ebp-2Ch]
  int *v15; // [esp+34h] [ebp-28h]
  int *v16; // [esp+38h] [ebp-24h]
  int *v17; // [esp+3Ch] [ebp-20h]
  __int16 v18; // [esp+40h] [ebp-1Ch] BYREF
  BOOL v19; // [esp+44h] [ebp-18h]
  __int16 v20[3]; // [esp+48h] [ebp-14h] BYREF
  bool v21; // [esp+4Eh] [ebp-Eh]
  bool v22; // [esp+4Fh] [ebp-Dh]
  int v23; // [esp+58h] [ebp-4h]

  v16 = cEncode::bCrucialEncode(a1, v11);
  v17 = cEncode::bCrucialEncode(dword_E39434, v12);
  v1 = fEncode::cCrucialEncode(v17);
  v19 = fEncode::cCrucialEncode(v16) != v1;
  v22 = v19;
  sub_E31E5B(v12);
  sub_E31E5B(v11);
  if ( !v22 )
    return 0;
  aCrucialEncode(v20, 0);
  aCrucialEncode(&v18, 7);
  while ( 1 )
  {
    v15 = cEncode::bCrucialEncode(dword_E39434, v10);
    v14 = v15;
    v23 = 0;
    v3 = fEncode::cCrucialEncode(v15);
    aCrucialEncode(&v13, v3);
    v21 = dCrucialEncode((unsigned __int16 *)v20, (unsigned __int16 *)&v13);
    v23 = -1;
    sub_E31E5B(v10);
    if ( !v21 )
      break;
    v4 = eCrucialEncode((unsigned __int16 *)v20);
```

```
    v5 = (_DWORD *)sub_E31984(a1, v4);
    v6 = sub_E31DCB(v5);
    v7 = (unsigned __int16)eCrucialEncode((unsigned __int16 *)&v18);
    v8 = eCrucialEncode((unsigned __int16 *)v20);
    v9 = (_DWORD *)sub_E31984(dword_E39434, v8);
    if ( v6 != (sub_E31DCB(v9) ^ v7) )
      return 0;
    sub_E31D73((unsigned __int16 *)v20);
    sub_E31D73((unsigned __int16 *)&v18);
  }
  return 1;
}
// E39434: using guessed type int dword_E39434[3];
// E31A6B: using guessed type int var_48[3];
// E31A6B: using guessed type int var_3C[3];
// E31A6B: using guessed type unsigned __int16 var_14[3];
// E31A6B: using guessed type int var_54[3];

//----- (00E31BC9) --------------------------------------------------------
int __cdecl main(int argc, const char **argv, const char **envp)
{
  _QWORD *v3; // eax
  __int64 v4; // rax
  _QWORD *v5; // eax
  int v7[4]; // [esp+0h] [ebp-34h] BYREF
  int v8[3]; // [esp+10h] [ebp-24h] BYREF
  __int64 v9; // [esp+1Ch] [ebp-18h] BYREF
  int v10; // [esp+24h] [ebp-10h] BYREF
  int v11; // [esp+30h] [ebp-4h]

  sub_E31CF9(v7);
  printf("%s()\n", "main");
  printf("Find correct password\n");
  while ( 1 )
  {
    aGetInput(v8);
    v11 = 0;
    if ( eEncode(v8) )
      break;
    v3 = sub_E31CC5(v7);
    v4 = sub_E31CA5(v3);
    printf("Incorrect!(%llu)\n", v4);
    v10 = 200;
    v5 = sub_E32927(&v9, &v10);
    sub_E32943(v5);
    v11 = -1;
    sub_E319DE(v8);
  }
  v11 = -1;
  sub_E319DE(v8);
  printf("Correct!\nPlease send DM with PW.\n");
  getchar();
  getchar();
  return 0;
}
```

```
// E31BC9: using guessed type int var_34[4];
// E31BC9: using guessed type int var_24[3];

//----- (00E31CA5) --------------------------------------------------------
__int64 __thiscall sub_401CA5(_QWORD *this)
{
  return *this ^ this[1];
}

//----- (00E31CC5) --------------------------------------------------------
_QWORD *__thiscall sub_401CC5(_QWORD *this)
{
  int *v1; // eax
  int *v2; // eax
  _DWORD *v3; // edi
  int v5[4]; // [esp+8h] [ebp-24h] BYREF
  int v6[4]; // [esp+18h] [ebp-14h] BYREF
  _QWORD *v7; // [esp+28h] [ebp-4h]

  v7 = this;
  v1 = sub_E321A2(v6, 1, 0);
  v2 = sub_E3216E(v7, v5, v1);
  v3 = v7;
  *(_DWORD *)v7 = *v2;
  *++v3 = v2[1];
  *++v3 = v2[2];
  v3[1] = v2[3];
  return v7;
}
// E31CC5: using guessed type int var_14[4];
// E31CC5: using guessed type int var_24[4];

//----- (00E31CF9) --------------------------------------------------------
int *__thiscall sub_401CF9(int *this)
{
  int v1; // edx
  int v2; // ecx

  *this = sub_E3296C();
  this[1] = v1;
  v2 = this[1];
  this[2] = *this;
  this[3] = v2;
  return this;
}
// E31D0A: variable ’v1’ is possibly undefined

//----- (00E31D23) --------------------------------------------------------
int __thiscall sub_401D23(unsigned __int16 *this)
{
  return *this ^ this[1];
}

//----- (00E31D3B) --------------------------------------------------------
bool __thiscall dCrucialEncode(unsigned __int16 *this, unsigned __int16 *a2)
```

```
{
  int v2; // esi

  v2 = (unsigned __int16)eCrucialEncode(this);
  return v2 < (unsigned __int16)eCrucialEncode(a2);
}

//----- (00E31D73) --------------------------------------------------------
unsigned __int16 *__thiscall sub_401D73(unsigned __int16 *this)
{
  __int16 *v1; // eax
  __int16 *v2; // eax
  __int16 v4; // [esp+0h] [ebp-Ch] BYREF
  __int16 v5; // [esp+4h] [ebp-8h] BYREF
  unsigned __int16 *v6; // [esp+8h] [ebp-4h]

  v6 = this;
  v1 = aCrucialEncode(&v5, 1);
  v2 = sub_E321D4(v6, &v4, (unsigned __int16 *)v1);
  *(_DWORD *)v6 = *(_DWORD *)v2;
  return v6;
}

//----- (00E31D9F) --------------------------------------------------------
__int16 *__thiscall sub_401D9F(__int16 *this, __int16 a2)
{
  *this = fCrucialEncode();
  this[1] = *this ^ a2;
  return this;
}

//----- (00E31DCB) --------------------------------------------------------
int __thiscall sub_401DCB(_DWORD *this)
{
  return *this ^ this[1];
}

//----- (00E31DDF) --------------------------------------------------------
int *__thiscall sub_401DDF(int *this, int a2)
{
  *this = sub_E329C4();
  this[1] = *this ^ a2;
  return this;
}

//----- (00E31E05) --------------------------------------------------------
int __thiscall sub_E31E05(_DWORD *this, int a2)
{
  return *this + 8 * a2;
}

//----- (00E31E22) --------------------------------------------------------
int __thiscall fEncode(_DWORD *this)
{
  return (this[1] - *this) >> 3;
```

```
}

//----- (00E31E40) --------------------------------------------------------
void __thiscall sub_E31E40(int *this, unsigned int a2)
{
  unsigned __int8 v2; // [esp+7h] [ebp-1h] BYREF

  sub_E329E8(this, a2, &v2);
}

//----- (00E31E5B) --------------------------------------------------------
int *__thiscall sub_E31E5B(int *this)
{
  return sub_E32216(this);
}

//----- (00E31E6C) --------------------------------------------------------
int *__thiscall dEncode(int *this, int *a2)
{
  int v4; // [esp+8h] [ebp-30h]
  int v5; // [esp+Ch] [ebp-2Ch]
  int *v6[2]; // [esp+10h] [ebp-28h] BYREF
  int *v7; // [esp+18h] [ebp-20h]
  int v8; // [esp+1Ch] [ebp-1Ch]
  int v9; // [esp+20h] [ebp-18h]
  int *v10; // [esp+24h] [ebp-14h]
  char v11[2]; // [esp+28h] [ebp-10h] BYREF
  char v12; // [esp+2Ah] [ebp-Eh] BYREF
  char v13; // [esp+2Bh] [ebp-Dh] BYREF
  int v14; // [esp+34h] [ebp-4h]

  v10 = this;
  kEncode(a2);
  v5 = MicrosoftVisualC14netruntime(&v12);
  LOBYTE(v4) = v11[1];
  qEncode(this, v4, v5);
  v7 = v10;
  v6[1] = a2;
  v8 = *a2;
  v9 = a2[1];
  bEncode(&v13, (int)v11, (int)v10);
  if ( v8 != v9 )
  {
    lEncode(v10, (v9 - v8) >> 3);
    v6[0] = v10;
    v14 = 0;
    v7[1] = rEncode(v10, v8, v9, *v7);
    v6[0] = 0;
    v14 = -1;
    jEncode(v6);
  }
  aEncode();
  return v10;
}
// E31EB6: variable 'v4' is possibly undefined
```

```
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E31F52) --------------------------------------------------------
_DWORD *__thiscall sub_E31F52(_DWORD *this)
{
  int v2; // [esp+0h] [ebp-14h]
  char v4; // [esp+13h] [ebp-1h] BYREF

  sub_E32B03(this, v2);
  sub_E3164B((int)&v4);
  return this;
}
// E31F67: variable 'v2' is possibly undefined

//----- (00E31F83) --------------------------------------------------------
_DWORD *__thiscall sub_401F83(_DWORD *this, _DWORD *a2)
{
  int v2; // eax

  v2 = a2[1];
  *this = *a2;
  this[1] = v2;
  return this;
}

//----- (00E31FA1) --------------------------------------------------------
_DWORD *__thiscall sub_401FA1(_DWORD *this, _DWORD *a2)
{
  int v2; // eax

  v2 = this[1];
  *a2 = *this;
  a2[1] = v2;
  return a2;
}

//----- (00E31FBF) --------------------------------------------------------
int __thiscall sub_401FBF(_DWORD *this)
{
  return this[4];
}

//----- (00E31FCE) --------------------------------------------------------
unsigned int *__thiscall sub_E31FCE(size_t *this, char a2)
{
  char v3; // [esp+0h] [ebp-14h]
  _DWORD *v4; // [esp+4h] [ebp-10h]
  size_t v5; // [esp+8h] [ebp-Ch]
  char v6; // [esp+13h] [ebp-1h] BYREF

  v5 = this[4];
  if ( v5 >= this[5] )
    return sub_E32B4B(this, 1u, v3, a2);
  this[4] = v5 + 1;
  v4 = sub_E324C0(this);
```

```
  nEncode((_BYTE *)v4 + v5, &a2);
  v6 = 0;
  return (unsigned int *)nEncode((_BYTE *)v4 + v5 + 1, &v6);
}
// E32039: variable 'v3' is possibly undefined

//----- (00E32042) --------------------------------------------------------
_BYTE *__stdcall sub_402042(_BYTE *a1, _BYTE *Src, size_t Size, char a4)
{
  char v5; // [esp+7h] [ebp-1h] BYREF

  sub_E32565(a1, Src, Size);
  nEncode(&a1[Size], &a4);
  v5 = 0;
  return nEncode(&a1[Size + 1], &v5);
}

//----- (00E3208B) --------------------------------------------------------
char *__thiscall sub_E3208B(_DWORD *this, int a2)
{
  return (char *)sub_E324C0(this) + a2;
}

//----- (00E320A1) --------------------------------------------------------
void __thiscall gEncode(void **this)
{
  mEncode(this);
  hEncode();
}

//----- (00E320BA) --------------------------------------------------------
unsigned int *__thiscall sub_E320BA(unsigned int *this, char *Src)
{
  int v3; // [esp+0h] [ebp-20h]
  char v5; // [esp+12h] [ebp-Eh] BYREF
  char v6; // [esp+13h] [ebp-Dh] BYREF
  int v7; // [esp+1Ch] [ebp-4h]

  sub_E32C45(this, v3);
  v7 = 0;
  bEncode(&v6, (int)&v5, (int)this);
  sub_E323C9((int)this);
  sub_E323FD(this, Src);
  aEncode();
  return this;
}
// E320E4: variable 'v3' is possibly undefined

//----- (00E32131) --------------------------------------------------------
void hEncode()
{
  iEncode();
}

//----- (00E32142) --------------------------------------------------------
```

```
void iEncode()
{
  aEncode();
}

//----- (00E32153) --------------------------------------------------------
int *__thiscall jEncode(int **this)
{
  int *result; // eax

  result = (int *)this;
  if ( *this )
    return sub_E32216(*this);
  return result;
}

//----- (00E3216E) --------------------------------------------------------
int *__thiscall sub_E3216E(_QWORD *this, int *a2, _QWORD *a3)
{
  __int64 v3; // kr00_8
  __int64 v4; // rax

  v3 = sub_E31CA5(this);
  v4 = sub_E31CA5(a3);
  sub_E321A2(a2, v4 + v3, (unsigned __int64)(v4 + v3) >> 32);
  return a2;
}

//----- (00E321A2) --------------------------------------------------------
int *__thiscall sub_4021A2(int *this, int a2, int a3)
{
  int v3; // edx
  int v4; // edx

  *this = sub_E3296C();
  this[1] = v3;
  v4 = this[1] ^ a3;
  this[2] = *this ^ a2;
  this[3] = v4;
  return this;
}
// E321B3: variable 'v3' is possibly undefined

//----- (00E321D4) --------------------------------------------------------
__int16 *__thiscall sub_E321D4(unsigned __int16 *this, __int16 *a2, unsigned __int16 *a3)
{
  __int16 v3; // si
  __int16 v4; // ax

  v3 = eCrucialEncode(this);
  v4 = eCrucialEncode(a3);
  aCrucialEncode(a2, v4 + v3);
  return a2;
}
```

```
//----- (00E32205) --------------------------------------------------------
void *__thiscall kEncode(void *this)
{
  return jCrucialEncode(this);
}

//----- (00E32216) --------------------------------------------------------
int *__thiscall sub_402216(int *this)
{
  int *result; // eax
  int *v2; // [esp+Ch] [ebp-20h]
  int *v3; // [esp+10h] [ebp-1Ch]

  v3 = this + 1;
  v2 = this + 2;
  aEncode();
  result = this;
  if ( *this )
  {
    sub_E325B7(this, *this, *v3);
    kEncode(this);
    sub_E3261E((void *)*this, (*v2 - *this) >> 3);
    *this = 0;
    *v3 = 0;
    result = v2;
    *v2 = 0;
  }
  return result;
}

//----- (00E322C8) --------------------------------------------------------
int __thiscall lEncode(int *this, unsigned int a2)
{
  int v2; // eax
  int result; // eax
  int *v4; // [esp+0h] [ebp-1Ch]
  int *v5; // [esp+4h] [ebp-18h]

  v5 = this + 1;
  v4 = this + 2;
  kEncode(this);
  v2 = oEncode(a2);
  *this = v2;
  *v5 = v2;
  result = v2 + 8 * a2;
  *v4 = result;
  return result;
}

//----- (00E3232A) --------------------------------------------------------
_BYTE *__thiscall mEncode(void **this)
{
  void *Block; // [esp+4h] [ebp-18h]
  char v4; // [esp+Fh] [ebp-Dh] BYREF
```

```
  aEncode();
  if ( pEncode(this) )
  {
    Block = *this;
    kEncode(this);
    sub_E32C63();
    sub_E327E0(Block, (int)this[5] + 1);
  }
  this[4] = 0;
  this[5] = (void *)15;
  v4 = 0;
  return nEncode(this, &v4);
}

//----- (00E323C9) --------------------------------------------------------
_BYTE *__thiscall sub_4023C9(int this)
{
  char v2; // [esp+7h] [ebp-1h] BYREF

  *(_DWORD *)(this + 16) = 0;
  *(_DWORD *)(this + 20) = 15;
  v2 = 0;
  return nEncode((_BYTE *)this, &v2);
}

//----- (00E323FD) --------------------------------------------------------
unsigned int *__thiscall sub_4023FD(unsigned int *this, char *Src)
{
  unsigned int v2; // eax
  size_t v3; // eax

  v2 = sub_E32533(Src);
  v3 = MicrosoftVisualC14netruntime(v2);
  return sub_E32424(this, Src, v3);
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32424) --------------------------------------------------------
unsigned int *__thiscall sub_402424(unsigned int *this, char *Src, size_t Size)
{
  char v4; // [esp+0h] [ebp-10h]
  char *v5; // [esp+4h] [ebp-Ch]
  char v7; // [esp+Fh] [ebp-1h] BYREF

  if ( Size > this[5] )
    return sub_E32C68(this, Size, v4, Src);
  v5 = (char *)sub_E324C0(this);
  this[4] = Size;
  sub_E326F2(v5, Src, Size);
  v7 = 0;
  nEncode(&v5[Size], &v7);
  return this;
}
// E32484: variable ’v4’ is possibly undefined
```

```
//----- (00E3248D) --------------------------------------------------------
_BYTE *__stdcall sub_40248D(_BYTE *a1, size_t Size, _BYTE *Src)
{
  char v4; // [esp+7h] [ebp-1h] BYREF

  sub_E32565(a1, Src, Size);
  v4 = 0;
  return nEncode(&a1[Size], &v4);
}

//----- (00E324C0) --------------------------------------------------------
_DWORD *__thiscall sub_E324C0(_DWORD *this)
{
  _DWORD *v2; // [esp+0h] [ebp-8h]

  v2 = this;
  if ( pEncode(this) )
    return (_DWORD *)MicrosoftVisualC14netruntime(*this);
  return v2;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E324F8) --------------------------------------------------------
bool __cdecl sub_4024F8(_DWORD *a1, _DWORD *a2)
{
  return *a1 == *a2;
}

//----- (00E32524) --------------------------------------------------------
_BYTE *__cdecl nEncode(_BYTE *a1, _BYTE *a2)
{
  _BYTE *result; // eax

  result = a1;
  *a1 = *a2;
  return result;
}

//----- (00E32533) --------------------------------------------------------
unsigned int __cdecl sub_E32533(const char *a1)
{
  return strlen(a1);
}

//----- (00E32565) --------------------------------------------------------
_BYTE *__cdecl sub_E32565(_BYTE *a1, _BYTE *Src, size_t Size)
{
  size_t i; // [esp+0h] [ebp-4h]

  if ( sub_E31328() )
  {
    for ( i = 0; i < Size; ++i )
      a1[i] = Src[i];
    return a1;
  }
```

```
  else
  {
    memcpy(a1, Src, Size);
    return a1;
  }
}

//----- (00E325B7) --------------------------------------------------------
void __thiscall sub_E325B7(void *this, int a2, int a3)
{
  kEncode(this);
  sub_E32C63();
}

//----- (00E325D9) --------------------------------------------------------
int __stdcall oEncode(unsigned int a1)
{
  int v1; // eax

  v1 = sub_E32D4A(a1);
  return sub_E32D70(v1);
}
// E32D70: using guessed type int __cdecl sub_402D70(_DWORD);

//----- (00E325F4) --------------------------------------------------------
_DWORD *__cdecl sub_4025F4(_DWORD *a1, _DWORD *a2)
{
  if ( *a1 >= *a2 )
    return a1;
  else
    return a2;
}

//----- (00E3261E) --------------------------------------------------------
void __stdcall sub_40261E(void *Block, int a2)
{
  sub_E32D9C(Block, 8 * a2);
}

//----- (00E3263A) --------------------------------------------------------
_BYTE *__thiscall sub_E3263A(_DWORD *this, int a2)
{
  _DWORD *v3; // [esp+4h] [ebp-Ch]
  char v5; // [esp+Fh] [ebp-1h] BYREF

  v5 = 0;
  v3 = sub_E324C0(this);
  this[4] = a2;
  return nEncode((_BYTE *)v3 + a2, &v5);
}

//----- (00E32677) --------------------------------------------------------
int __thiscall sub_E32677(void *this)
{
  void *v1; // eax
```

```
  int v3; // [esp+0h] [ebp-18h] BYREF
  int v4; // [esp+4h] [ebp-14h] BYREF
  int v5; // [esp+8h] [ebp-10h]
  int v6; // [esp+Ch] [ebp-Ch] BYREF
  int v7[2]; // [esp+10h] [ebp-8h] BYREF

  v7[1] = (int)this;
  v1 = kEncode(this);
  v6 = std::numeric_limits<unsigned int>::max(v1);
  v7[0] = 16;
  v5 = *sub_E325F4(&v6, v7);
  v4 = v5 - 1;
  v3 = unknown_libname_1();
  return *sub_E32B21(&v3, &v4);
}
// E3159B: using guessed type int unknown_libname_1(void);
// E324F0: using guessed type int __cdecl std::numeric_limits<unsigned int>::max(_DWORD);

//----- (00E326CF) --------------------------------------------------------
bool __thiscall pEncode(_DWORD *this)
{
  return this[5] >= 0x10u;
}

//----- (00E326F2) --------------------------------------------------------
char *__cdecl sub_E326F2(char *a1, char *Src, size_t Size)
{
  char *i; // [esp+4h] [ebp-10h]
  size_t k; // [esp+8h] [ebp-Ch]
  size_t j; // [esp+Ch] [ebp-8h]
  char v7; // [esp+13h] [ebp-1h]

  if ( sub_E31328() )
  {
    if ( a1 == Src )
    {
      return a1;
    }
    else
    {
      v7 = 1;
      for ( i = Src; i != &Src[Size]; ++i )
      {
        if ( a1 == i )
        {
          v7 = 0;
          break;
        }
      }
      if ( v7 )
      {
        for ( j = 0; j < Size; ++j )
          a1[j] = Src[j];
      }
      else
```

```
      {
        for ( k = 0; k < Size; ++k )
          a1[Size - 1 - k] = Src[Size - 1 - k];
      }
      return a1;
    }
  }
  else
  {
    memmove(a1, Src, Size);
    return a1;
  }
}

//----- (00E327C5) --------------------------------------------------------
int __stdcall sub_4027C5(int a1)
{
  int v1; // eax

  v1 = sub_E32DEA(a1);
  return sub_E32D70(v1);
}
// E32D70: using guessed type int __cdecl sub_402D70(_DWORD);

//----- (00E327E0) --------------------------------------------------------
void __stdcall sub_4027E0(void *Block, int a2)
{
  sub_E32D9C(Block, a2);
}

//----- (00E327F8) --------------------------------------------------------
void __thiscall sub_4027F8(_DWORD *this, unsigned int a2)
{
  if ( this[4] < a2 )
    sub_E32813();
}

//----- (00E32813) --------------------------------------------------------
void sub_402813()
{
  std::_Xout_of_range("invalid string position");
}
// E37078: using guessed type void __cdecl std::_Xout_of_range(const char *);

//----- (00E32823) --------------------------------------------------------
struct std::_Facet_base *__cdecl sub_402823(_DWORD *a1)
{
  struct std::_Facet_base *v2; // [esp+0h] [ebp-28h]
  char v3[4]; // [esp+4h] [ebp-24h] BYREF
  unsigned int v4; // [esp+8h] [ebp-20h]
  int (__thiscall ***v5)(_DWORD, int); // [esp+Ch] [ebp-1Ch] BYREF
  struct std::_Facet_base *v6; // [esp+10h] [ebp-18h]
  struct std::_Facet_base *v7; // [esp+14h] [ebp-14h]
  struct std::_Facet_base *v8; // [esp+18h] [ebp-10h] BYREF
  int v9; // [esp+24h] [ebp-4h]
```

```
  std::_Lockit::_Lockit((std::_Lockit *)v3, 0);
  v9 = 0;
  v8 = (struct std::_Facet_base *)dword_E39430;
  v4 = std::locale::id::operator unsigned int(std::ctype<char>::id);
  v6 = (struct std::_Facet_base *)sub_E31721(a1, v4);
  if ( !v6 )
  {
    if ( v8 )
    {
      v6 = v8;
    }
    else
    {
      if ( std::ctype<char>::_Getcat(&v8, a1) == -1 )
        sub_E31698();
      v7 = v8;
      sub_E331DB(&v5, (char)v8);
      LOBYTE(v9) = 1;
      std::_Facet_Register(v7);
      (*(void (__thiscall **)(struct std::_Facet_base *))(*(_DWORD *)v7 + 4))(v7);
      dword_E39430 = (int)v8;
      v6 = v8;
      sub_E32DF7((int *)&v5);
      LOBYTE(v9) = 0;
      sub_E32E13(&v5);
    }
  }
  v2 = v6;
  v9 = -1;
  std::_Lockit::~_Lockit((std::_Lockit *)v3);
  return v2;
}
// E3704C: using guessed type int __cdecl std::ctype<char>::_Getcat(_DWORD, _DWORD);
// E37054: using guessed type int __thiscall std::locale::id::operator unsigned int(_DWORD);
// E37084: using guessed type _DWORD __thiscall std::_Lockit::_Lockit(std::_Lockit *__hidden this, _DW
ORD);
// E37088: using guessed type void __thiscall std::_Lockit::~_Lockit(std::_Lockit *__hidden this);
// E39430: using guessed type int dword_409430;
// E32823: using guessed type char var_24[4];

//----- (00E3290E) --------------------------------------------------------
int __cdecl sub_40290E(int a1, _DWORD *a2)
{
  int v2; // eax

  v2 = MicrosoftVisualC14netruntime(a1);
  return sub_E32F66(v2, a2);
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32927) --------------------------------------------------------
_QWORD *__thiscall sub_402927(_QWORD *this, int *a2)
{
  *this = *a2;
```

```
  return this;
}

//----- (00E32943) --------------------------------------------------------
BOOL __cdecl sub_402943(void *a1)
{
  _DWORD *v1; // eax
  _DWORD *v2; // eax
  int v4; // [esp+0h] [ebp-10h] BYREF
  int v5; // [esp+8h] [ebp-8h] BYREF

  v1 = sub_E317C3(&v5);
  v2 = sub_E33200(&v4, v1, a1);
  return sub_E33230(v2);
}

//----- (00E3296C) --------------------------------------------------------
int sub_40296C()
{
  __int64 v1; // [esp+0h] [ebp-10h]
  __int64 v2; // [esp+8h] [ebp-8h]

  v2 = sub_E315BF();
  v1 = sub_E315B6();
  return sub_E33285(v1, SHIDWORD(v1), v2, SHIDWORD(v2));
}

//----- (00E3299E) --------------------------------------------------------
__int16 sub_40299E()
{
  __int16 v0; // ax
  __int16 v2; // [esp+4h] [ebp-4h]

  v2 = MicrosoftVisualC14netruntime2();
  v0 = gCrucialEncode();
  return hCrucialEncode(v0, v2);
}
// E315A5: using guessed type int __scrt_stub_for_initialize_mta(void);
// E315AC: using guessed type int unknown_libname_2(void);

//----- (00E329C4) --------------------------------------------------------
int sub_4029C4()
{
  int v1; // [esp+0h] [ebp-8h]
  int v2; // [esp+4h] [ebp-4h]

  v2 = unknown_libname_1();
  v1 = std::numeric_limits<int>::min();
  return sub_E3331B(v1, v2);
}
// E31591: using guessed type int std::numeric_limits<int>::min(void);
// E3159B: using guessed type int unknown_libname_1(void);

//----- (00E329E8) --------------------------------------------------------
void __thiscall sub_E329E8(int *this, unsigned int a2, unsigned __int8 *a3)
```

```
{
  int v3; // [esp+4h] [ebp-1Ch]
  unsigned int v4; // [esp+8h] [ebp-18h]
  int v5; // [esp+14h] [ebp-Ch]
  int *v7; // [esp+1Ch] [ebp-4h]

  v7 = this + 1;
  v4 = (this[1] - *this) >> 3;
  if ( a2 >= v4 )
  {
    if ( a2 > v4 )
    {
      if ( a2 <= (this[2] - *this) >> 3 )
      {
        v3 = *v7;
        *v7 = sub_E32E6F(this, *v7, a2 - v4, *a3);
        sub_E32E64(v3, v3);
      }
      else
      {
        sub_E33363(this, a2, a3);
      }
    }
  }
  else
  {
    v5 = *this + 8 * a2;
    sub_E32E64(v5, *v7);
    sub_E325B7(this, v5, *v7);
    *v7 = v5;
  }
}

//----- (00E32ABF) --------------------------------------------------------
_DWORD *__thiscall qEncode(_DWORD *this, int a2, int a3)
{
  MicrosoftVisualC14netruntime(a3);
  sEncode(this);
  return this;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32ADE) --------------------------------------------------------
int __thiscall rEncode(void *this, int a2, int a3, int a4)
{
  void *v4; // eax

  v4 = kEncode(this);
  return tEncode(a2, a3, a4, (int)v4);
}

//----- (00E32B03) --------------------------------------------------------
_DWORD *__thiscall sub_E32B03(_DWORD *this, int a2)
{
  jCrucialEncode(this);
```

```
  sEncode(this);
  return this;
}

//----- (00E32B21) --------------------------------------------------------
_DWORD *__cdecl sub_402B21(_DWORD *a1, _DWORD *a2)
{
  if ( *a2 >= *a1 )
    return a1;
  else
    return a2;
}

//----- (00E32B4B) --------------------------------------------------------
unsigned int *__thiscall sub_E32B4B(size_t *this, size_t a2, char a3, char a4)
{
  _BYTE *v4; // eax
  _BYTE *v5; // eax
  _BYTE *v7; // [esp+0h] [ebp-28h]
  void *Block; // [esp+8h] [ebp-20h]
  unsigned int v9; // [esp+Ch] [ebp-1Ch]
  unsigned int v10; // [esp+10h] [ebp-18h]
  int v11; // [esp+18h] [ebp-10h] BYREF
  size_t Size; // [esp+1Ch] [ebp-Ch]
  unsigned int *v13; // [esp+20h] [ebp-8h]
  void *Src; // [esp+24h] [ebp-4h]

  v13 = this;
  Src = this;
  Size = this[4];
  if ( sub_E32677(this) - Size < a2 )
    sub_E31664();
  v9 = *((_DWORD *)Src + 5);
  v10 = sub_E32E91(v13, a2 + Size);
  kEncode(v13);
  v11 = sub_E327C5(v10 + 1);
  aEncode();
  *((_DWORD *)Src + 4) = a2 + Size;
  *((_DWORD *)Src + 5) = v10;
  v4 = (_BYTE *)MicrosoftVisualC14netruntime(v11);
  v7 = v4;
  if ( v9 < 0x10 )
  {
    sub_E32042(v4, Src, Size, a4);
    sub_E32D1E((int)Src, (int)&v11);
  }
  else
  {
    Block = *(void **)Src;
    v5 = (_BYTE *)MicrosoftVisualC14netruntime(*(_DWORD *)Src);
    sub_E32042(v7, v5, Size, a4);
    sub_E327E0(Block, v9 + 1);
    *(_DWORD *)Src = v11;
  }
  return v13;
```

```
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32C45) --------------------------------------------------------
_DWORD *__thiscall sub_E32C45(_DWORD *this, int a2)
{
  jCrucialEncode(this);
  sub_E32EC3(this);
  return this;
}

//----- (00E32C63) --------------------------------------------------------
void sub_402C63()
{
  ;
}

//----- (00E32C68) --------------------------------------------------------
unsigned int *__thiscall sub_E32C68(unsigned int *this, size_t Size, char a3, _BYTE *Src)
{
  _BYTE *v4; // eax
  unsigned int v6; // [esp+4h] [ebp-10h]
  unsigned int v7; // [esp+8h] [ebp-Ch]
  int v8; // [esp+Ch] [ebp-8h] BYREF
  unsigned int *v9; // [esp+10h] [ebp-4h]

  v9 = this;
  if ( Size > sub_E32677(this) )
    sub_E31664();
  v6 = v9[5];
  v7 = sub_E32E91(v9, Size);
  kEncode(v9);
  v8 = sub_E327C5(v7 + 1);
  aEncode();
  v9[4] = Size;
  v9[5] = v7;
  v4 = (_BYTE *)MicrosoftVisualC14netruntime(v8);
  sub_E3248D(v4, Size, Src);
  if ( v6 < 0x10 )
  {
    sub_E32D1E((int)v9, (int)&v8);
  }
  else
  {
    sub_E327E0((void *)*v9, v6 + 1);
    *v9 = v8;
  }
  return v9;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32D1E) --------------------------------------------------------
int __cdecl sub_402D1E(int a1, int a2)
{
  void *v2; // eax
```

```
  int result; // eax
  _DWORD *v4; // [esp+0h] [ebp-4h]

  v2 = (void *)MicrosoftVisualC14netruntime(a1);
  v4 = xEncode(4u, v2);
  result = *(_DWORD *)MicrosoftVisualC14netruntime(a2);
  *v4 = result;
  return result;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E32D4A) --------------------------------------------------------
unsigned int __cdecl sub_402D4A(unsigned int a1)
{
  if ( a1 > 0x1FFFFFFF )
    sub_E3152F();
  return 8 * a1;
}

//----- (00E32D70) --------------------------------------------------------
#error "E32D90: call analysis failed (funcsize=17)"

//----- (00E32D9C) --------------------------------------------------------
void __cdecl sub_402D9C(void *Block, unsigned int a2)
{
  if ( a2 >= 0x1000 )
    sub_E315D8(&Block, &a2);
  sub_E35312(Block);
}

//----- (00E32DEA) --------------------------------------------------------
int __cdecl sub_402DEA(int a1)
{
  return a1;
}

//----- (00E32DF7) --------------------------------------------------------
int __thiscall sub_402DF7(int *this)
{
  int v2; // [esp+4h] [ebp-4h] BYREF

  v2 = 0;
  return sub_E33591(this, &v2);
}

//----- (00E32E13) --------------------------------------------------------
int (__thiscall ***__thiscall sub_402E13(int (__thiscall ****this)(_DWORD, int)))(_DWORD, int)
{
  int (__thiscall ***result)(_DWORD, int); // eax

  result = (int (__thiscall ***)(_DWORD, int))this;
  if ( *this )
  {
    jCrucialEncode(this);
    return sub_E32EE5(*this);
```

```
  }
  return result;
}

//----- (00E32E44) --------------------------------------------------------
_DWORD *__thiscall sEncode(_DWORD *this)
{
  *this = 0;
  this[1] = 0;
  this[2] = 0;
  return this;
}

//----- (00E32E64) --------------------------------------------------------
void __stdcall sub_402E64(int a1, int a2)
{
  ;
}

//----- (00E32E6F) --------------------------------------------------------
int __thiscall sub_E32E6F(void *this, int a2, int a3, int a4)
{
  void *v4; // eax

  v4 = kEncode(this);
  return sub_E335AC(a2, a3, (int)v4);
}

//----- (00E32E91) --------------------------------------------------------
unsigned int __thiscall sub_E32E91(unsigned int *this, int a2)
{
  int v3; // [esp+4h] [ebp-8h]

  v3 = sub_E32677(this);
  return sub_E32F19(a2, this[5], v3);
}

//----- (00E32EC3) --------------------------------------------------------
_DWORD *__thiscall sub_E32EC3(_DWORD *this)
{
  jCrucialEncode(this);
  this[4] = 0;
  this[5] = 0;
  return this;
}

//----- (00E32EE5) --------------------------------------------------------
int (__thiscall ***__stdcall sub_402EE5(int (__thiscall ***a1)(_DWORD, int)))(_DWORD, int)
{
  int (__thiscall ***result)(_DWORD, int); // eax

  result = a1;
  if ( a1 )
    return (int (__thiscall ***)(_DWORD, int))(**a1)(a1, 1);
  return result;
```

```
}

//----- (00E32F19) --------------------------------------------------------
unsigned int __cdecl sub_402F19(int a1, unsigned int a2, unsigned int a3)
{
  int v4; // [esp+0h] [ebp-8h] BYREF
  int v5; // [esp+4h] [ebp-4h] BYREF

  v5 = a1 | 0xF;
  if ( (a1 | 0xFu) > a3 )
    return a3;
  if ( a2 > a3 - (a2 >> 1) )
    return a3;
  v4 = a2 + (a2 >> 1);
  return *sub_E325F4(&v5, &v4);
}

//----- (00E32F66) --------------------------------------------------------
int __cdecl sub_E32F66(int a1, _DWORD *a2)
{
  unsigned int v2; // eax
  unsigned __int8 v3; // al
  char v4; // al
  int v6; // [esp+0h] [ebp-88h] BYREF
  int v7; // [esp+4h] [ebp-84h]
  int v8; // [esp+8h] [ebp-80h]
  _DWORD *v9[2]; // [esp+Ch] [ebp-7Ch] BYREF
  int v10[2]; // [esp+14h] [ebp-74h] BYREF
  __int64 v11; // [esp+1Ch] [ebp-6Ch]
  __int64 v12; // [esp+24h] [ebp-64h]
  __int64 v13; // [esp+2Ch] [ebp-5Ch]
  int v14; // [esp+34h] [ebp-54h]
  std::ios_base *v15; // [esp+3Ch] [ebp-4Ch]
  int v16; // [esp+40h] [ebp-48h]
  struct std::_Facet_base *v17; // [esp+44h] [ebp-44h]
  int v18; // [esp+48h] [ebp-40h] BYREF
  int v19; // [esp+4Ch] [ebp-3Ch]
  int v20; // [esp+50h] [ebp-38h]
  int v21; // [esp+54h] [ebp-34h]
  int v22; // [esp+58h] [ebp-30h]
  _DWORD *v23; // [esp+5Ch] [ebp-2Ch]
  _DWORD *v24; // [esp+60h] [ebp-28h]
  int v25; // [esp+64h] [ebp-24h]
  int v26; // [esp+68h] [ebp-20h] BYREF
  int v27; // [esp+6Ch] [ebp-1Ch]
  int v28; // [esp+70h] [ebp-18h]
  int v29; // [esp+74h] [ebp-14h]
  int *v30; // [esp+78h] [ebp-10h]
  int v31; // [esp+7Ch] [ebp-Ch]
  int v32; // [esp+80h] [ebp-8h]
  int v33; // [esp+84h] [ebp-4h]

  v30 = &v6;
  v28 = 0;
  HIBYTE(v29) = 0;
```

```
sub_E336F6(v10, a1, 0);
v33 = 0;
if ( sub_E336E7(v10) )
{
  v25 = *(_DWORD *)(*(_DWORD *)a1 + 4) + a1;
  v24 = (_DWORD *)std::ios_base::getloc(v25, v9);
  v23 = v24;
  LOBYTE(v33) = 1;
  v17 = sub_E32823(v24);
  LOBYTE(v33) = 0;
  sub_E316D8(v9);
  sub_E3392C(a2, 0);
  LOBYTE(v33) = 2;
  v13 = std::ios_base::width((std::ios_base *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1));
  if ( v13 <= 0
    || (v12 = std::ios_base::width((std::ios_base *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1)),
        v2 = sub_E32677(a2),
        (unsigned int)v12 >= v2) )
  {
    v27 = sub_E32677(a2);
  }
  else
  {
    v11 = std::ios_base::width((std::ios_base *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1));
    v27 = v11;
  }
  v22 = std::ios::rdbuf(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1, v6, v7);
  v21 = std::streambuf::sgetc(v22);
  v26 = v21;
  while ( v27 )
  {
    v18 = std::numeric_limits<unsigned int>::max(v6);
    if ( sub_E324F8(&v18, &v26) )
    {
      v28 |= 1u;
      break;
    }
    v3 = std::_Narrow_char_traits<char,int>::to_char_type((int)&v26);
    BYTE2(v29) = std::ctype<char>::is(v17, 72, v3);
    if ( BYTE2(v29) )
      break;
    v4 = std::_Narrow_char_traits<char,int>::to_char_type((int)&v26);
    sub_E31FCE(a2, v4);
    HIBYTE(v29) = 1;
    --v27;
    v20 = std::ios::rdbuf(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1, v6, v7);
    v19 = std::streambuf::snextc(v20);
    v26 = v19;
  }
  v33 = 0;
}
v15 = (std::ios_base *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + a1);
std::ios_base::width(v15, 0i64);
if ( !HIBYTE(v29) )
  v28 |= 2u;
```

```
  std::ios::setstate(
    *(_DWORD *)(*(_DWORD *)a1 + 4) + a1,
    v28,
    0,
    v6,
    v7,
    v8,
    v9[0],
    v9[1],
    v10[0],
    v10[1],
    v11,
    HIDWORD(v11),
    v12,
    HIDWORD(v12),
    v13,
    HIDWORD(v13),
    v14,
    *(_DWORD *)(*(_DWORD *)a1 + 4) + a1,
    v15,
    v16,
    v17,
    v18,
    v19,
    v20,
    v21,
    v22,
    v23,
    v24,
    v25,
    v26,
    v27,
    v28,
    v29,
    v30,
    v31,
    v32,
    v33);
  v14 = a1;
  v33 = -1;
  sub_E33618(v10);
  return v14;
}
// E324F0: using guessed type int __cdecl std::numeric_limits<unsigned int>::max(_DWORD);
// E37034: using guessed type int __thiscall std::ios::setstate(_DWORD, _DWORD, _DWORD, _DWORD,
  _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _D
WORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWO
RD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD,
  _DWORD, _DWORD, _DWORD, _DWORD, _DWORD);
// E37038: using guessed type int __thiscall std::ios::rdbuf(_DWORD, _DWORD, _DWORD);
// E3703C: using guessed type int __thiscall std::streambuf::sgetc(_DWORD);
// E37040: using guessed type int __thiscall std::ios_base::getloc(_DWORD, _DWORD);
// E37044: using guessed type int __thiscall std::streambuf::snextc(_DWORD);
// E37048: using guessed type __int64 __thiscall std::ios_base::width(std::ios_base *__hidden this);
// E37050: using guessed type int __thiscall std::ctype<char>::is(_DWORD, _DWORD, _DWORD);
```

```
// E3708C: using guessed type __int64 __thiscall std::ios_base::width(std::ios_base *__hidden this, __int6
4);
// E32F66: using guessed type _DWORD var_74[2];

//----- (00E331DB) --------------------------------------------------------
_DWORD *__thiscall sub_E331DB(_DWORD *this, char a2)
{
  int v3; // [esp+0h] [ebp-Ch]

  sub_E33BAD(this, v3, (int)&a2);
  return this;
}
// E331F4: variable 'v3' is possibly undefined

//----- (00E33200) --------------------------------------------------------
_DWORD *__cdecl sub_403200(_DWORD *a1, _DWORD *a2, void *a3)
{
  int *v3; // eax
  _DWORD *v4; // eax
  int v6; // [esp+0h] [ebp-10h] BYREF
  int v7; // [esp+8h] [ebp-8h] BYREF

  v3 = sub_E31FA1(a2, &v7);
  v4 = sub_E33BCB(&v6, v3, a3);
  sub_E31F83(a1, v4);
  return a1;
}

//----- (00E33230) --------------------------------------------------------
BOOL __cdecl sub_403230(_DWORD *a1)
{
  BOOL result; // eax
  _DWORD *v2; // eax
  xtime v3; // [esp+0h] [ebp-20h] BYREF
  int v4; // [esp+10h] [ebp-10h] BYREF
  int v5[2]; // [esp+18h] [ebp-8h] BYREF

  while ( 1 )
  {
    sub_E317C3(v5);
    result = sub_E33C1C(a1, v5);
    if ( result )
      break;
    v2 = sub_E33C46(&v4, a1, v5);
    sub_E33C80((int)&v3, v2);
    Thrd_sleep(&v3);
  }
  return result;
}
// E33230: using guessed type _DWORD var_8[2];

//----- (00E33285) --------------------------------------------------------
int __cdecl sub_403285(int a1, int a2, int a3, int a4)
{
  unsigned int v4; // eax
```

```
  int v6[1250]; // [esp+0h] [ebp-139Ch] BYREF
  int v7[4]; // [esp+1388h] [ebp-14h] BYREF
  char v8; // [esp+139Bh] [ebp-1h] BYREF

  jCrucialEncode(&v8);
  v4 = kCrucialEncode();
  sub_E3390C(v6, v4);
  sub_E336C5(v7, a1, a2, a3, a4);
  return sub_E33D4C(v7, (int)v6);
}
// E33285: using guessed type _DWORD var_139C[1250];
// E33285: using guessed type int var_14[4];

//----- (00E332D3) --------------------------------------------------------
__int16 __cdecl sub_4032D3(__int16 a1, __int16 a2)
{
  unsigned int v2; // eax
  int v4[1250]; // [esp+0h] [ebp-1390h] BYREF
  __int16 v5[3]; // [esp+1388h] [ebp-8h] BYREF
  char v6; // [esp+138Fh] [ebp-1h] BYREF

  jCrucialEncode(&v6);
  v2 = kCrucialEncode();
  sub_E3390C(v4, v2);
  lCrucialEncode(v5, a1, a2);
  return mCrucialEncode(v5, (int)v4);
}
// E332D3: using guessed type _DWORD var_1390[1250];
// E332D3: using guessed type __int16 var_8[3];

//----- (00E3331B) --------------------------------------------------------
int __cdecl sub_40331B(int a1, int a2)
{
  unsigned int v2; // eax
  int v4[1250]; // [esp+0h] [ebp-1394h] BYREF
  int v5[2]; // [esp+1388h] [ebp-Ch] BYREF
  char v6; // [esp+1393h] [ebp-1h] BYREF

  jCrucialEncode(&v6);
  v2 = kCrucialEncode();
  sub_E3390C(v4, v2);
  sub_E3368D(v5, a1, a2);
  return sub_E33D98(v5, (int)v4);
}
// E3331B: using guessed type _DWORD var_1394[1250];

//----- (00E33363) --------------------------------------------------------
int __thiscall sub_E33363(int *this, unsigned int a2, unsigned __int8 *a3)
{
  int v4; // [esp+0h] [ebp-4Ch] BYREF
  int v5; // [esp+10h] [ebp-3Ch]
  void *v6; // [esp+14h] [ebp-38h]
  int v7; // [esp+18h] [ebp-34h]
  int *v8; // [esp+1Ch] [ebp-30h]
  int *v9; // [esp+20h] [ebp-2Ch]
```

```
  int v10; // [esp+24h] [ebp-28h]
  int *v11; // [esp+28h] [ebp-24h]
  int v12; // [esp+2Ch] [ebp-20h]
  int v13; // [esp+30h] [ebp-1Ch]
  void *Block; // [esp+34h] [ebp-18h]
  int *v15; // [esp+38h] [ebp-14h]
  int *v16; // [esp+3Ch] [ebp-10h]
  int v17; // [esp+48h] [ebp-4h]

  v16 = &v4;
  v15 = this;
  if ( a2 > sub_E338D6(this) )
    sub_E337AD();
  v11 = v15;
  v8 = v15;
  v9 = v15 + 1;
  v10 = (v15[1] - *v15) >> 3;
  v12 = sub_E3385F(v15, a2);
  v6 = kEncode(v15);
  Block = (void *)oEncode(v12);
  v13 = (int)Block + 8 * v10;
  v7 = v13;
  v17 = 0;
  v5 = sub_E32E6F(v15, v13, a2 - v10, *a3);
  v7 = v5;
  sub_E338B1(v15, *v8, *v9, (int)Block);
  v17 = -1;
  return sub_E337BD((void **)v15, Block, a2, v12);
}

//----- (00E3349C) --------------------------------------------------------
int __cdecl tEncode(int a1, int a2, int a3, int a4)
{
  int v5[3]; // [esp+0h] [ebp-20h] BYREF
  int v6; // [esp+Ch] [ebp-14h]
  int v7; // [esp+10h] [ebp-10h]
  int v8; // [esp+1Ch] [ebp-4h]

  v7 = unknown_libname_4(&a1);
  v6 = unknown_libname_4(&a2);
  sub_E33665(v5, a3, a4);
  v8 = 0;
  while ( v7 != v6 )
  {
    uEncode(v5, v7);
    v7 += 8;
  }
  a3 = sub_E33629(v5);
  v8 = -1;
  sub_E33643();
  return a3;
}
// E33529: using guessed type _DWORD __cdecl unknown_libname_4(_DWORD);
// E3349C: using guessed type _DWORD var_20[3];
```

```
//----- (00E33533) --------------------------------------------------------
unsigned int __cdecl sub_E33533(unsigned int a1)
{
  void *v1; // eax
  int v3; // [esp+0h] [ebp-Ch]
  int v4; // [esp+8h] [ebp-4h]

  if ( a1 + 35 <= a1 )
    sub_E3152F();
  v1 = operator new(a1 + 35, v3, (const char *)(a1 + 35), v4);
  if ( !v1 )
    invalid_parameter_noinfo_noreturn();
  *(_DWORD *)((((unsigned int)v1 + 35) & 0xFFFFFFE0) - 4) = v1;
  return ((unsigned int)v1 + 35) & 0xFFFFFFE0;
}
// E33552: variable 'v3' is possibly undefined
// E33552: variable 'v4' is possibly undefined

//----- (00E33591) --------------------------------------------------------
int __cdecl sub_403591(int *a1, int *a2)
{
  int v3; // [esp+0h] [ebp-4h]

  v3 = *a1;
  *a1 = *a2;
  return v3;
}

//----- (00E335AC) --------------------------------------------------------
int __cdecl sub_E335AC(int a1, int a2, int a3)
{
  int v4[3]; // [esp+0h] [ebp-1Ch] BYREF
  int v5; // [esp+Ch] [ebp-10h]
  int v6; // [esp+18h] [ebp-4h]

  sub_E33665(v4, a1, a3);
  v6 = 0;
  while ( a2 )
  {
    sub_E33E0A(v4);
    --a2;
  }
  v5 = sub_E33629(v4);
  v6 = -1;
  sub_E33643();
  return v5;
}
// E335AC: using guessed type _DWORD var_1C[3];

//----- (00E33618) --------------------------------------------------------
int __thiscall sub_E33618(_DWORD *this)
{
  return sub_E33751(this);
}
```

```
//----- (00E33629) --------------------------------------------------------
int __thiscall sub_403629(_DWORD *this)
{
  *this = this[1];
  return this[1];
}

//----- (00E33643) --------------------------------------------------------
void sub_403643()
{
  sub_E32C63();
}

//----- (00E33665) --------------------------------------------------------
_DWORD *__thiscall sub_403665(_DWORD *this, int a2, int a3)
{
  *this = a2;
  this[1] = a2;
  this[2] = a3;
  return this;
}

//----- (00E3368D) --------------------------------------------------------
_DWORD *__thiscall sub_E3368D(_DWORD *this, int a2, int a3)
{
  sub_E33957(this, a2, a3);
  return this;
}

//----- (00E336A9) --------------------------------------------------------
_WORD *__thiscall lCrucialEncode(_WORD *this, __int16 a2, __int16 a3)
{
  nCrucialEncode(this, a2, a3);
  return this;
}

//----- (00E336C5) --------------------------------------------------------
_DWORD *__thiscall sub_E336C5(_DWORD *this, int a2, int a3, int a4, int a5)
{
  sub_E3399D(this, a2, a3, a4, a5);
  return this;
}

//----- (00E336E7) --------------------------------------------------------
char __thiscall sub_4036E7(_BYTE *this)
{
  return this[4];
}

//----- (00E336F6) --------------------------------------------------------
_DWORD *__thiscall sub_E336F6(_DWORD *this, int a2, int a3)
{
  sub_E339C6(this, a2);
  *((_BYTE *)this + 4) = std::istream::_Ipfx(*this, a3);
  return this;
```

```
}
// E37058: using guessed type int __thiscall std::istream::_Ipfx(_DWORD, _DWORD);

//----- (00E33751) --------------------------------------------------------
// positive sp value has been detected, the output may be wrong!
int __thiscall sub_E33751(_DWORD *this)
{
  int result; // eax

  result = ((int (__thiscall *)(int))std::ios::rdbuf)(*(_DWORD *)(*(_DWORD *)*this + 4) + *this);
  if ( result )
    return (*(int (__thiscall **)(int, _DWORD *))(*(_DWORD *)result + 8))(result, this);
  return result;
}
// E337A7: positive sp value 8 has been found
// E37038: using guessed type int __thiscall std::ios::rdbuf(_DWORD, _DWORD, _DWORD);

//----- (00E337AD) --------------------------------------------------------
void __noreturn sub_4037AD()
{
  std::_Xlength_error("vector too long");
}
// E3706C: using guessed type void __cdecl __noreturn std::_Xlength_error(const char *);

//----- (00E337BD) --------------------------------------------------------
char *__thiscall sub_E337BD(void **this, char *a2, int a3, int a4)
{
  char *result; // eax
  void **v5; // [esp+Ch] [ebp-14h]
  int *v6; // [esp+10h] [ebp-10h]

  v6 = (int *)(this + 1);
  v5 = this + 2;
  aEncode();
  if ( *this )
  {
    sub_E325B7(this, (int)*this, *v6);
    kEncode(this);
    sub_E3261E(*this, ((_BYTE *)*v5 - (_BYTE *)*this) >> 3);
  }
  *this = a2;
  *v6 = (int)&a2[8 * a3];
  result = &a2[8 * a4];
  *v5 = result;
  return result;
}

//----- (00E3385F) --------------------------------------------------------
int __thiscall sub_E3385F(_DWORD *this, unsigned int a2)
{
  int v3; // [esp+4h] [ebp-Ch]
  unsigned int v5; // [esp+Ch] [ebp-4h]

  v5 = sub_E33A30(this);
  v3 = sub_E338D6(this);
```

```
  if ( v5 > v3 - (v5 >> 1) )
    return v3;
  if ( v5 + (v5 >> 1) >= a2 )
    return v5 + (v5 >> 1);
  return a2;
}

//----- (00E338B1) --------------------------------------------------------
int __thiscall sub_E338B1(void *this, int a2, int a3, int a4)
{
  int v5; // [esp+4h] [ebp-4h]

  LOBYTE(v5) = 0;
  return sub_E33A0B(this, a2, a3, a4, v5);
}
// E338CD: variable 'v5' is possibly undefined

//----- (00E338D6) --------------------------------------------------------
int __thiscall sub_E338D6(void *this)
{
  int v2; // [esp+0h] [ebp-Ch] BYREF
  int v3[2]; // [esp+4h] [ebp-8h] BYREF

  v3[1] = (int)this;
  kEncode(this);
  v3[0] = sub_E33A4E();
  v2 = unknown_libname_1();
  return *sub_E32B21(&v2, v3);
}
// E3159B: using guessed type int unknown_libname_1(void);

//----- (00E3390C) --------------------------------------------------------
_DWORD *__thiscall sub_E3390C(_DWORD *this, unsigned int a2)
{
  sub_E33A58(this, a2, -1, 1812433253);
  return this;
}

//----- (00E3392C) --------------------------------------------------------
_DWORD *__thiscall sub_E3392C(_DWORD *this, unsigned int a2)
{
  sub_E327F8(this, a2);
  sub_E3263A(this, a2);
  return this;
}

//----- (00E33957) --------------------------------------------------------
_DWORD *__thiscall sub_E33957(_DWORD *this, int a2, int a3)
{
  sub_E33A80(this, a2, a3);
  return this;
}

//----- (00E3397A) --------------------------------------------------------
_WORD *__thiscall sub_E3397A(_WORD *this, __int16 a2, __int16 a3)
```

```
{
  oCrucialEncode(this, a2, a3);
  return this;
}

//----- (00E3399D) --------------------------------------------------------
_DWORD *__thiscall sub_E3399D(_DWORD *this, int a2, int a3, int a4, int a5)
{
  sub_E33AB8(this, a2, a3, a4, a5);
  return this;
}

//----- (00E339C6) --------------------------------------------------------
int __thiscall sub_4039C6(_DWORD *this, int a2)
{
  int v3; // [esp+4h] [ebp-8h]
  int v4; // [esp+8h] [ebp-4h]

  *this = a2;
  v4 = std::ios::rdbuf(*(_DWORD *)(*(_DWORD *)*this + 4) + *this, *this, this);
  if ( v4 )
    (*(void (__thiscall **)(int))(*(_DWORD *)v4 + 4))(v4);
  return v3;
}
// E33A04: variable 'v3' is possibly undefined
// E37038: using guessed type int __thiscall std::ios::rdbuf(_DWORD, _DWORD, _DWORD);

//----- (00E33A0B) --------------------------------------------------------
int __thiscall sub_E33A0B(void *this, int a2, int a3, int a4, int a5)
{
  void *v5; // eax

  v5 = kEncode(this);
  return tEncode(a2, a3, a4, (int)v5);
}

//----- (00E33A30) --------------------------------------------------------
int __thiscall sub_403A30(_DWORD *this)
{
  return (this[2] - *this) >> 3;
}

//----- (00E33A4E) --------------------------------------------------------
int sub_403A4E()
{
  return 0x1FFFFFFF;
}

//----- (00E33A58) --------------------------------------------------------
_DWORD *__thiscall sub_E33A58(_DWORD *this, unsigned int a2, int a3, int a4)
{
  this[1249] = a3;
  sub_E33ADA(this, a2, a4);
  return this;
}
```

```
//----- (00E33A80) --------------------------------------------------------
_DWORD *__thiscall sub_E33A80(_DWORD *this, int a2, int a3)
{
  sub_E33B49(this, a2, a3);
  return this;
}

//----- (00E33A9C) --------------------------------------------------------
_WORD *__thiscall oCrucialEncode(_WORD *this, __int16 a2, __int16 a3)
{
  pCrucialEncode(this, a2, a3);
  return this;
}

//----- (00E33AB8) --------------------------------------------------------
_DWORD *__thiscall sub_E33AB8(_DWORD *this, int a2, int a3, int a4, int a5)
{
  sub_E33B85(this, a2, a3, a4, a5);
  return this;
}

//----- (00E33ADA) --------------------------------------------------------
_DWORD *__thiscall sub_E33ADA(_DWORD *this, unsigned int a2, int a3)
{
  int v3; // eax
  _DWORD *result; // eax
  unsigned int v5; // [esp+8h] [ebp-8h]
  unsigned int i; // [esp+Ch] [ebp-4h]

  this[1] = a2;
  v5 = a2;
  for ( i = 1; i < 0x270; ++i )
  {
    v3 = a3 * (v5 ^ (v5 >> 30));
    this[i + 1] = v3 + i;
    v5 = v3 + i;
  }
  result = this;
  *this = 624;
  return result;
}

//----- (00E33B49) --------------------------------------------------------
_DWORD *__thiscall sub_403B49(_DWORD *this, int a2, int a3)
{
  _DWORD *result; // eax

  *this = a2;
  result = this;
  this[1] = a3;
  return result;
}

//----- (00E33B65) --------------------------------------------------------
```

```
_WORD *__thiscall sub_403B65(_WORD *this, __int16 a2, __int16 a3)
{
  _WORD *result; // eax

  *this = a2;
  result = this;
  this[1] = a3;
  return result;
}

//----- (00E33B85) ----------------------------------------------------------
_DWORD *__thiscall sub_403B85(_DWORD *this, int a2, int a3, int a4, int a5)
{
  _DWORD *result; // eax

  *this = a2;
  this[1] = a3;
  result = this;
  this[2] = a4;
  this[3] = a5;
  return result;
}

//----- (00E33BAD) ----------------------------------------------------------
_DWORD *__thiscall sub_403BAD(_DWORD *this, int a2, int a3)
{
  *this = *(_DWORD *)MicrosoftVisualC14netruntime(a3);
  return this;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E33BCB) ----------------------------------------------------------
_DWORD *__cdecl sub_403BCB(_DWORD *a1, int *a2, void *a3)
{
  int v3; // eax
  __int64 v4; // kr00_8
  __int64 *v5; // eax
  __int64 v7; // [esp+8h] [ebp-18h] BYREF
  __int64 v8; // [esp+10h] [ebp-10h] BYREF
  int v9[2]; // [esp+18h] [ebp-8h] BYREF

  v3 = a2[1];
  v9[0] = *a2;
  v9[1] = v3;
  v4 = sub_E31882(v9);
  v5 = sub_E33F7F(&v7, a3);
  v8 = sub_E31882(v5) + v4;
  sub_E31864(a1, &v8);
  return a1;
}

//----- (00E33C1C) ----------------------------------------------------------
bool __cdecl sub_403C1C(_DWORD *a1, _DWORD *a2)
{
  return !sub_E33FAC(a2, a1);
```

```
}

//----- (00E33C46) --------------------------------------------------------
_DWORD *__cdecl sub_403C46(_DWORD *a1, _DWORD *a2, _DWORD *a3)
{
  int v4; // [esp+0h] [ebp-18h] BYREF
  int v5; // [esp+8h] [ebp-10h] BYREF
  int *v6; // [esp+10h] [ebp-8h]
  int *v7; // [esp+14h] [ebp-4h]

  v7 = sub_E31FA1(a3, &v5);
  v6 = sub_E31FA1(a2, &v4);
  sub_E33FDF(a1, v6, v7);
  return a1;
}

//----- (00E33C80) --------------------------------------------------------
bool __cdecl sub_403C80(int a1, void *a2)
{
  _DWORD *v2; // eax
  _QWORD *v3; // eax
  int v5; // [esp+0h] [ebp-48h] BYREF
  int v6; // [esp+8h] [ebp-40h] BYREF
  int v7; // [esp+10h] [ebp-38h] BYREF
  __int64 v8; // [esp+18h] [ebp-30h] BYREF
  __int64 v9; // [esp+20h] [ebp-28h] BYREF
  int v10[2]; // [esp+28h] [ebp-20h] BYREF
  __int64 v11; // [esp+30h] [ebp-18h] BYREF
  __int64 v12; // [esp+38h] [ebp-10h] BYREF
  _DWORD *v13; // [esp+40h] [ebp-8h]
  bool v14; // [esp+47h] [ebp-1h]

  v11 = 864000000000000i64;
  v9 = 0x412A5E0000000000i64;
  v13 = sub_E31797(&v7);
  v2 = sub_E31FA1(v13, &v6);
  sub_E34034(&v12, v2);
  v14 = sub_E34061(&v9, a2);
  if ( v14 )
  {
    sub_E33E6F(&v12, &v11);
  }
  else
  {
    v3 = sub_E340BD(&v5, a2);
    sub_E33E6F(&v12, v3);
  }
  sub_E33E95(v10, &v12);
  *(_QWORD *)a1 = sub_E31882(v10);
  sub_E33F52(&v8, v10);
  sub_E33E49(&v12, &v8);
  *(_DWORD *)(a1 + 8) = sub_E31882(&v12);
  return v14;
}
// E33C80: using guessed type _DWORD var_20[2];
```

```
//----- (00E33D4C) --------------------------------------------------------
int __thiscall sub_E33D4C(int *this, int a2)
{
  return sub_E3415C(this, a2, *this, this[1], this[2], this[3]);
}

//----- (00E33D73) --------------------------------------------------------
__int16 __thiscall mCrucialEncode(__int16 *this, int a2)
{
  return qCrucialEncode(this, a2, *this, this[1]);
}

//----- (00E33D98) --------------------------------------------------------
int __thiscall sub_E33D98(int *this, int a2)
{
  return sub_E34278(this, a2, *this, this[1]);
}

//----- (00E33DB9) --------------------------------------------------------
int __thiscall uEncode(_DWORD *this, int a2)
{
  int result; // eax
  void *v3; // [esp+4h] [ebp-Ch]
  int v4; // [esp+8h] [ebp-8h]

  v4 = MicrosoftVisualC14netruntime(a2);
  v3 = (void *)MicrosoftVisualC14netruntime(this[1]);
  vEncode(this[2], v3, v4);
  result = this[1] + 8;
  this[1] = result;
  return result;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E33E0A) --------------------------------------------------------
int __thiscall sub_403E0A(_DWORD *this)
{
  int result; // eax
  void *v2; // [esp+4h] [ebp-8h]

  v2 = (void *)MicrosoftVisualC14netruntime(this[1]);
  sub_E3430D(this[2], v2);
  result = this[1] + 8;
  this[1] = result;
  return result;
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E33E49) --------------------------------------------------------
_QWORD *__thiscall sub_403E49(_QWORD *this, _QWORD *a2)
{
  *this -= *a2;
  return this;
}
```

```
//----- (00E33E6F) --------------------------------------------------------
_QWORD *__thiscall sub_403E6F(_QWORD *this, _QWORD *a2)
{
  *this += *a2;
  return this;
}

//----- (00E33E95) --------------------------------------------------------
_DWORD *__cdecl sub_403E95(_DWORD *a1, void *a2)
{
  __int64 v3; // [esp+8h] [ebp-1Ch] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 1;
  v4 = 0;
  v3 = sub_E31882(a2) / 1000000000;
  sub_E31864(a1, &v3);
  return a1;
}

//----- (00E33F52) --------------------------------------------------------
__int64 *__thiscall sub_403F52(__int64 *this, void *a2)
{
  _DWORD *v2; // eax
  __int64 v3; // rax
  int v5; // [esp+0h] [ebp-Ch] BYREF
  __int64 *v6; // [esp+8h] [ebp-4h]

  v6 = this;
  v2 = sub_E34B5C(&v5, a2);
  v3 = sub_E31882(v2);
  *v6 = v3;
  return v6;
}

//----- (00E33F7F) --------------------------------------------------------
__int64 *__thiscall sub_403F7F(__int64 *this, void *a2)
{
  _DWORD *v2; // eax
  __int64 v3; // rax
  int v5; // [esp+0h] [ebp-Ch] BYREF
  __int64 *v6; // [esp+8h] [ebp-4h]

  v6 = this;
  v2 = sub_E34C18(&v5, a2);
  v3 = sub_E31882(v2);
  *v6 = v3;
  return v6;
}

//----- (00E33FAC) --------------------------------------------------------
bool __cdecl sub_403FAC(_DWORD *a1, _DWORD *a2)
{
```

```
  int v3; // [esp+0h] [ebp-18h] BYREF
  int v4; // [esp+8h] [ebp-10h] BYREF
  int *v5; // [esp+10h] [ebp-8h]
  int *v6; // [esp+14h] [ebp-4h]

  v6 = sub_E31FA1(a2, &v4);
  v5 = sub_E31FA1(a1, &v3);
  return sub_E34CD4(v5, v6);
}

//----- (00E33FDF) --------------------------------------------------------
_DWORD *__cdecl sub_403FDF(_DWORD *a1, int *a2, int *a3)
{
  int v3; // eax
  int v4; // eax
  __int64 v5; // rax
  __int64 v7; // [esp+8h] [ebp-18h] BYREF
  int v8[2]; // [esp+10h] [ebp-10h] BYREF
  int v9[2]; // [esp+18h] [ebp-8h] BYREF

  v3 = a2[1];
  v9[0] = *a2;
  v9[1] = v3;
  v4 = a3[1];
  v8[0] = *a3;
  v8[1] = v4;
  v5 = sub_E31882(v9);
  v7 = v5 - sub_E31882(v8);
  sub_E31864(a1, &v7);
  return a1;
}

//----- (00E34034) --------------------------------------------------------
__int64 *__thiscall sub_404034(__int64 *this, void *a2)
{
  _DWORD *v2; // eax
  __int64 v3; // rax
  int v5; // [esp+0h] [ebp-Ch] BYREF
  __int64 *v6; // [esp+8h] [ebp-4h]

  v6 = this;
  v2 = sub_E34D3E(&v5, a2);
  v3 = sub_E31882(v2);
  *v6 = v3;
  return v6;
}

//----- (00E34061) --------------------------------------------------------
bool __cdecl sub_404061(void *a1, void *a2)
{
  double *v2; // eax
  double *v3; // eax
  double v5; // [esp+0h] [ebp-2Ch] BYREF
  double v6; // [esp+8h] [ebp-24h] BYREF
  double v7; // [esp+10h] [ebp-1Ch]
```

```
  double v8; // [esp+18h] [ebp-14h]
  double v9; // [esp+20h] [ebp-Ch]

  v2 = sub_E34DF1(&v6, a1);
  v9 = sub_E31893(v2);
  v7 = v9;
  v3 = sub_E34E1B(&v5, a2);
  v8 = sub_E31893(v3);
  return v8 > v7;
}
// E34061: using guessed type double var_24;
// E34061: using guessed type double var_2C;

//----- (00E340BD) --------------------------------------------------------
_DWORD *__cdecl sub_4040BD(_DWORD *a1, void *a2)
{
  __int64 v3; // [esp+18h] [ebp-Ch] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 1;
  v4 = 1;
  v3 = sub_E31882(a2);
  sub_E31864(a1, &v3);
  return a1;
}
// E340BD: using guessed type __int64 var_C;

//----- (00E3415C) --------------------------------------------------------
int __thiscall sub_40415C(void *this, int a2, int a3, int a4, int a5, int a6)
{
  __int64 v6; // rax
  __int64 v7; // rax
  int v9[4]; // [esp+0h] [ebp-34h] BYREF
  __int64 v10; // [esp+10h] [ebp-24h]
  unsigned __int64 v11; // [esp+18h] [ebp-1Ch]
  __int64 v12; // [esp+20h] [ebp-14h]
  __int64 v13; // [esp+28h] [ebp-Ch]
  void *v14; // [esp+30h] [ebp-4h]

  v14 = this;
  sub_E34700(v9, a2);
  LODWORD(v6) = sub_E347C1(a3, a4);
  v13 = v6;
  LODWORD(v7) = sub_E347C1(a5, a6);
  v12 = v7;
  v10 = v7 - v13;
  if ( (HIDWORD(v10) & (unsigned int)v10) == -1 )
    v11 = sub_E34528((int)v9);
  else
    v11 = sub_E3459A((int)v9, v12 - v13 + 1);
  return sub_E347C1(v13 + v11, (v13 + v11) >> 32);
}
// E3417D: variable 'v6' is possibly undefined
// E34190: variable 'v7' is possibly undefined
```

```
// E3415C: using guessed type _DWORD var_34[4];

//----- (00E341FE) --------------------------------------------------------
__int16 __thiscall qCrucialEncode(void *this, int a2, __int16 a3, __int16 a4)
{
  _DWORD v5[4]; // [esp+0h] [ebp-1Ch] BYREF
  __int16 v6; // [esp+10h] [ebp-Ch]
  unsigned __int16 v7; // [esp+14h] [ebp-8h]
  unsigned __int16 v8; // [esp+18h] [ebp-4h]

  v5[3] = this;
  rCrucialEncode(v5, a2);
  v8 = sCrucialEncode(a3);
  v7 = sCrucialEncode(a4);
  if ( v7 - v8 == 0xFFFF )
    v6 = tCrucialEncode(v5);
  else
    v6 = sub_E34478(v5, v7 - v8 + 1);
  return sCrucialEncode(v8 + v6);
}

//----- (00E34278) --------------------------------------------------------
int __thiscall sub_E34278(void *this, int a2, int a3, int a4)
{
  _DWORD v5[4]; // [esp+0h] [ebp-1Ch] BYREF
  unsigned int v6; // [esp+10h] [ebp-Ch]
  int v7; // [esp+14h] [ebp-8h]
  int v8; // [esp+18h] [ebp-4h]

  v5[3] = this;
  rCrucialEncode(v5, a2);
  v8 = sub_E3478F(a3);
  v7 = sub_E3478F(a4);
  if ( v7 - v8 == -1 )
    v6 = tCrucialEncode(v5);
  else
    v6 = sub_E3437B(v5, v7 - v8 + 1);
  return sub_E3478F(v8 + v6);
}

//----- (00E342E0) --------------------------------------------------------
_DWORD *__cdecl vEncode(int a1, void *a2, int a3)
{
  _DWORD *v4; // [esp+0h] [ebp-8h]
  _DWORD *v5; // [esp+4h] [ebp-4h]

  v4 = xEncode(8u, a2);
  v5 = (_DWORD *)MicrosoftVisualC14netruntime(a3);
  return wEncode(v4, v5);
}
// E3163A: using guessed type _DWORD __cdecl MicrosoftVisualC14netruntime(_DWORD);

//----- (00E3430D) --------------------------------------------------------
int *__cdecl sub_40430D(int a1, void *a2)
{
```

```
  int *v3; // [esp+0h] [ebp-4h]

  v3 = (int *)xEncode(8u, a2);
  return sub_E34802(v3);
}

//----- (00E3432A) --------------------------------------------------------
unsigned int __thiscall tCrucialEncode(_DWORD *this)
{
  unsigned int i; // [esp+4h] [ebp-8h]
  unsigned int v4; // [esp+8h] [ebp-4h]
  int v5; // [esp+8h] [ebp-4h]

  v4 = 0;
  for ( i = 0; i < 0x20; i += this[1] )
  {
    v5 = 2 * (v4 << (this[1] - 1));
    v4 = v5 | sub_E34823(this);
  }
  return v4;
}

//----- (00E3437B) --------------------------------------------------------
unsigned int __thiscall sub_E3437B(_DWORD *this, unsigned int a2)
{
  unsigned int v4; // [esp+4h] [ebp-8h]
  int v5; // [esp+4h] [ebp-8h]
  unsigned int i; // [esp+8h] [ebp-4h]

  do
  {
    v4 = 0;
    for ( i = 0; i < a2 - 1; i = this[2] | (2 * (i << (this[1] - 1))) )
    {
      v5 = 2 * (v4 << (this[1] - 1));
      v4 = v5 | sub_E34823(this);
    }
  }
  while ( v4 / a2 >= i / a2 && i % a2 != a2 - 1 );
  return v4 % a2;
}

//----- (00E3441A) --------------------------------------------------------
_DWORD *__thiscall rCrucialEncode(_DWORD *this, int a2)
{
  int v2; // esi
  int v4; // [esp+0h] [ebp-8h]

  *this = a2;
  this[1] = 32;
  for ( this[2] = -1; ; this[2] >>= 1 )
  {
    v2 = std::numeric_limits<unsigned int>::max(v4);
    if ( (unsigned int)(v2 - gCrucialEncode()) >= this[2] )
      break;
```

```
    --this[1];
  }
  return this;
}
// E3444B: variable 'v4' is possibly undefined
// E315A5: using guessed type int gCrucialEncode(void);
// E324F0: using guessed type int __cdecl std::numeric_limits<unsigned int>::max(_DWORD);

//----- (00E34478) --------------------------------------------------------
__int16 __thiscall sub_E34478(_DWORD *this, unsigned __int16 a2)
{
  unsigned int v4; // [esp+8h] [ebp-8h]
  int v5; // [esp+8h] [ebp-8h]
  unsigned int i; // [esp+Ch] [ebp-4h]

  do
  {
    v4 = 0;
    for ( i = 0; i < (unsigned int)a2 - 1; i = this[2] | (2 * (i << (this[1] - 1))) )
    {
      v5 = 2 * (v4 << (this[1] - 1));
      v4 = v5 | sub_E34823(this);
    }
  }
  while ( v4 / a2 >= i / a2 && i % a2 != a2 - 1 );
  return v4 % a2;
}

//----- (00E34528) --------------------------------------------------------
__int64 __thiscall sub_E34528(int this)
{
  __int64 v2; // [esp+0h] [ebp-10h]
  __int64 v3; // [esp+0h] [ebp-10h]
  unsigned int i; // [esp+Ch] [ebp-4h]

  v2 = 0i64;
  for ( i = 0; i < 0x40; i += *(_DWORD *)(this + 4) )
  {
    v3 = 2 * (v2 << ((unsigned __int8)*(_DWORD *)(this + 4) - 1));
    v2 = v3 | sub_E34857((_QWORD *)this);
  }
  return v2;
}

//----- (00E3459A) --------------------------------------------------------
unsigned __int64 __thiscall sub_E3459A(int this, unsigned __int64 a2)
{
  unsigned __int64 v3; // [esp+30h] [ebp-14h]
  __int64 v4; // [esp+30h] [ebp-14h]
  unsigned __int64 i; // [esp+38h] [ebp-Ch]

  do
  {
    v3 = 0i64;
    for ( i = 0i64; i < a2 - 1; i = *(_QWORD *)(this + 8) | (2 * (i << ((unsigned __int8)*(_DWORD *)(this + 4) -
```

```
 1))) )
    {
      v4 = 2 * (v3 << ((unsigned __int8)*(_DWORD *)(this + 4) - 1));
      v3 = v4 | sub_E34857((_QWORD *)this);
    }
  }
  while ( v3 / a2 >= i / a2 && i % a2 != a2 - 1 );
  return v3 % a2;
}

//----- (00E34700) --------------------------------------------------------
_DWORD *__thiscall sub_E34700(_DWORD *this, int a2)
{
  int v2; // esi
  int v4; // [esp+0h] [ebp-14h]

  *this = a2;
  this[1] = 64;
  this[2] = -1;
  this[3] = -1;
  while ( 1 )
  {
    v2 = std::numeric_limits<unsigned int>::max(v4);
    if ( (unsigned __int64)(unsigned int)(v2 - gCrucialEncode()) >= *((_QWORD *)this + 1) )
      break;
    --this[1];
    *((_QWORD *)this + 1) >>= 1;
  }
  return this;
}
// E34744: variable 'v4' is possibly undefined
// E315A5: using guessed type int gCrucialEncode(void);
// E324F0: using guessed type int __cdecl std::numeric_limits<unsigned int>::max(_DWORD);

//----- (00E3478F) --------------------------------------------------------
int __cdecl sub_40478F(int a1)
{
  return sub_E348A7(a1);
}

//----- (00E347A8) --------------------------------------------------------
__int16 __cdecl sub_4047A8(__int16 a1)
{
  return uCrucialEncode(a1);
}

//----- (00E347C1) --------------------------------------------------------
int __cdecl sub_4047C1(int a1, int a2)
{
  return operator"" _l(a1, a2);
}
// E348DA: using guessed type _DWORD __cdecl operator"" _l(_DWORD, _DWORD);

//----- (00E347DE) --------------------------------------------------------
_DWORD *__thiscall wEncode(_DWORD *this, _DWORD *a2)
```

```
{
  *this = *a2;
  this[1] = a2[1];
  return this;
}

//----- (00E34802) --------------------------------------------------------
int *__thiscall sub_404802(int *this)
{
  *this = sub_E329C4();
  this[1] = *this;
  return this;
}

//----- (00E34823) --------------------------------------------------------
unsigned int __thiscall sub_404823(_DWORD *this)
{
  unsigned int v1; // esi
  unsigned int v3; // [esp+4h] [ebp-8h]

  do
  {
    v1 = sub_E348E5((_DWORD *)*this);
    v3 = v1 - gCrucialEncode();
  }
  while ( v3 > this[2] );
  return v3;
}
// E315A5: using guessed type int __scrt_stub_for_initialize_mta(void);

//----- (00E34857) --------------------------------------------------------
__int64 __thiscall sub_404857(_QWORD *this)
{
  unsigned int v1; // esi
  __int64 v3; // [esp+4h] [ebp-10h]

  do
  {
    v1 = sub_E348E5(*(_DWORD **)this);
    v3 = v1 - gCrucialEncode();
  }
  while ( (unsigned __int64)(unsigned int)v3 > this[1] );
  return v3;
}
// E315A5: using guessed type int __scrt_stub_for_initialize_mta(void);

//----- (00E348A7) --------------------------------------------------------
int __cdecl sub_4048A7(int a1)
{
  return a1 + 0x80000000;
}

//----- (00E348D1) --------------------------------------------------------
__int16 __cdecl sub_4048D1(__int16 a1)
{
```

```
  return a1;
}

//----- (00E348E5) --------------------------------------------------------
int __thiscall sub_E348E5(_DWORD *this)
{
  unsigned int v2; // [esp+0h] [ebp-Ch]
  int v4; // [esp+8h] [ebp-4h]
  unsigned int v5; // [esp+8h] [ebp-4h]

  if ( *this == 624 )
  {
    sub_E3497E(this);
  }
  else if ( *this >= 0x4E0u )
  {
    sub_E34A02(this);
  }
  v2 = this[++*this];
  v4 = v2 ^ this[1249] & (v2 >> 11);
  v5 = v4 ^ (v4 << 7) & 0x9D2C5680 ^ ((v4 ^ (v4 << 7) & 0x9D2C5680) << 15) & 0xEFC60000;
  return v5 ^ (v5 >> 18);
}

//----- (00E3497E) --------------------------------------------------------
unsigned int __thiscall sub_E3497E(_DWORD *this)
{
  unsigned int result; // eax
  int v2; // [esp+0h] [ebp-10h]
  unsigned int i; // [esp+Ch] [ebp-4h]

  for ( i = 624; i < 0x4E0; ++i )
  {
    if ( (this[i - 622] & 1) != 0 )
      v2 = -1727483681;
    else
      v2 = 0;
    this[i + 1] = this[i - 226] ^ v2 ^ ((this[i - 622] & 0x7FFFFFFF | this[i - 623] & 0x80000000) >> 1);
    result = i + 1;
  }
  return result;
}

//----- (00E34A02) --------------------------------------------------------
_BYTE *__thiscall sub_E34A02(_BYTE *this)
{
  _BYTE *result; // eax
  int v2; // [esp+0h] [ebp-20h]
  int v3; // [esp+8h] [ebp-18h]
  int v4; // [esp+10h] [ebp-10h]
  unsigned int i; // [esp+1Ch] [ebp-4h]

  for ( i = 0; i < 0xE3; ++i )
  {
    if ( (this[4 * i + 2504] & 1) != 0 )
```

```
      v4 = -1727483681;
    else
      v4 = 0;
    *(_DWORD *)&this[4 * i + 4] = *(_DWORD *)&this[4 * i + 4088] ^ v4 ^ ((*(_DWORD *)&this[4 * i + 2504]
& 0x7FFFFFFF | *(_DWORD *)&this[4 * i + 2500] & 0x80000000) >> 1);
  }
  while ( i < 0x26F )
  {
    if ( (this[4 * i + 2504] & 1) != 0 )
      v3 = -1727483681;
    else
      v3 = 0;
    *(_DWORD *)&this[4 * i + 4] = *(_DWORD *)&this[4 * i - 904] ^ v3 ^ ((*(_DWORD *)&this[4 * i + 2504] &
 0x7FFFFFFF | *(_DWORD *)&this[4 * i + 2500] & 0x80000000) >> 1);
    ++i;
  }
  if ( (this[4] & 1) != 0 )
    v2 = -1727483681;
  else
    v2 = 0;
  *(_DWORD *)&this[4 * i + 4] = *((_DWORD *)this + 397) ^ v2 ^ ((*((_DWORD *)this + 1) & 0x7FFFFFFF |
 *(_DWORD *)&this[4 * i + 2500] & 0x80000000) >> 1);
  result = this;
  *(_DWORD *)this = 0;
  return result;
}

//----- (00E34B5C) ----------------------------------------------------------
_DWORD *__cdecl sub_404B5C(_DWORD *a1, void *a2)
{
  __int64 v3; // [esp+10h] [ebp-14h] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 0;
  v4 = 1;
  v3 = 1000000000 * sub_E31882(a2);
  sub_E31864(a1, &v3);
  return a1;
}

//----- (00E34C18) ----------------------------------------------------------
_DWORD *__cdecl sub_404C18(_DWORD *a1, void *a2)
{
  __int64 v3; // [esp+10h] [ebp-14h] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 0;
  v4 = 1;
  v3 = 1000000 * sub_E31882(a2);
  sub_E31864(a1, &v3);
  return a1;
}
```

```
//----- (00E34CD4) --------------------------------------------------------
bool __cdecl sub_404CD4(int *a1, int *a2)
{
  int v2; // eax
  int v3; // eax
  __int64 v4; // kr00_8
  int v6[2]; // [esp+8h] [ebp-24h] BYREF
  int v7[2]; // [esp+10h] [ebp-1Ch] BYREF

  v2 = a1[1];
  v7[0] = *a1;
  v7[1] = v2;
  v3 = a2[1];
  v6[0] = *a2;
  v6[1] = v3;
  v4 = sub_E31882(v7);
  return v4 < sub_E31882(v6);
}

//----- (00E34D3E) --------------------------------------------------------
_DWORD *__cdecl sub_404D3E(_DWORD *a1, void *a2)
{
  __int64 v3; // [esp+10h] [ebp-14h] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 0;
  v4 = 1;
  v3 = 100 * sub_E31882(a2);
  sub_E31864(a1, &v3);
  return a1;
}

//----- (00E34DF1) --------------------------------------------------------
double *__thiscall sub_404DF1(double *this, void *a2)
{
  _QWORD *v2; // eax
  double v3; // st7
  __int64 v5; // [esp+0h] [ebp-Ch] BYREF
  double *v6; // [esp+8h] [ebp-4h]

  v6 = this;
  v2 = sub_E34E45(&v5, a2);
  v3 = sub_E31893(v2);
  *v6 = v3;
  return v6;
}

//----- (00E34E1B) --------------------------------------------------------
double *__thiscall sub_404E1B(double *this, void *a2)
{
  _QWORD *v2; // eax
  double v3; // st7
  __int64 v5; // [esp+0h] [ebp-Ch] BYREF
  double *v6; // [esp+8h] [ebp-4h]
```

```
  v6 = this;
  v2 = sub_E34F17(&v5, a2);
  v3 = sub_E31893(v2);
  *v6 = v3;
  return v6;
}

//----- (00E34E45) --------------------------------------------------------
_QWORD *__cdecl sub_404E45(_QWORD *a1, void *a2)
{
  double v3; // [esp+20h] [ebp-1Ch] BYREF
  double v4; // [esp+28h] [ebp-14h]
  char v5; // [esp+3Ah] [ebp-2h]
  char v6; // [esp+3Bh] [ebp-1h]

  v6 = 0;
  v5 = 1;
  v4 = sub_E31893(a2);
  v3 = v4 * 1000000000.0;
  sub_E34FF1(a1, &v3);
  return a1;
}

//----- (00E34F17) --------------------------------------------------------
_QWORD *__cdecl sub_404F17(_QWORD *a1, void *a2)
{
  double v3; // [esp+18h] [ebp-Ch] BYREF
  char v4; // [esp+22h] [ebp-2h]
  char v5; // [esp+23h] [ebp-1h]

  v5 = 1;
  v4 = 1;
  v3 = (double)sub_E31882(a2);
  sub_E34FF1(a1, &v3);
  return a1;
}
// E34F17: using guessed type double var_C;

//----- (00E34FF1) --------------------------------------------------------
_QWORD *__thiscall sub_404FF1(_QWORD *this, _QWORD *a2)
{
  *this = *a2;
  return this;
}

//----- (00E35312) --------------------------------------------------------
void __cdecl sub_405312(void *Block)
{
  j_free(Block);
}

//----- (00E35320) --------------------------------------------------------
_DWORD *__thiscall sub_405320(_DWORD *Block, char a2)
{
```

```
  *Block = &type_info::‘vftable’;
  if ( (a2 & 1) != 0 )
    sub_E35312(Block);
  return Block;
}
// E371B4: using guessed type void *type_info::‘vftable’;

//----- (00E353EE) --------------------------------------------------------
int sub_4053EE()
{
  __scrt_initialize_default_local_stdio_options();
  return 0;
}
// E35B00: using guessed type int __scrt_initialize_default_local_stdio_options(void);

//----- (00E353F6) --------------------------------------------------------
int sub_4053F6()
{
  int v0; // eax

  sub_E3592E();
  v0 = UserMathErrorFunction();
  return set_new_mode(v0);
}

//----- (00E35597) --------------------------------------------------------
_DWORD *__thiscall sub_405597(_DWORD *this)
{
  _DWORD *result; // eax

  this[1] = 0;
  result = this;
  this[2] = 0;
  this[1] = "bad allocation";
  *this = &std::bad_alloc::‘vftable’;
  return result;
}
// E371C8: using guessed type void *std::bad_alloc::‘vftable’;

//----- (00E355AF) --------------------------------------------------------
void __noreturn sub_4055AF()
{
  int pExceptionObject[3]; // [esp+0h] [ebp-Ch] BYREF

  sub_E35597(pExceptionObject);
  CxxThrowException(pExceptionObject, (_ThrowInfo *)&_TI2_AVbad_alloc_std__);
}
// E355AF: using guessed type void __noreturn sub_4055AF();
// E355AF: using guessed type _DWORD pExceptionObject[3];

//----- (00E355CC) --------------------------------------------------------
void __noreturn sub_4055CC()
{
  int pExceptionObject[3]; // [esp+0h] [ebp-Ch] BYREF
```

```
  sub_E314D2(pExceptionObject);
  CxxThrowException(pExceptionObject, (_ThrowInfo *)&_TI3_AVbad_array_new_length_std__);
}
// E355CC: using guessed type void __noreturn sub_4055CC();
// E355CC: using guessed type _DWORD pExceptionObject[3];

//----- (00E357B9) --------------------------------------------------------
int sub_4057B9()
{
  return 1;
}

//----- (00E358E8) --------------------------------------------------------
int __cdecl UserMathErrorFunction()
{
  return 0;
}

//----- (00E3592E) --------------------------------------------------------
LPTOP_LEVEL_EXCEPTION_FILTER sub_40592E()
{
  return SetUnhandledExceptionFilter(__scrt_unhandled_exception_filter);
}

//----- (00E35990) --------------------------------------------------------
void sub_405990()
{
  dword_E390F0 = 0;
}
// E390F0: using guessed type int dword_4090F0;

//----- (00E35ACA) --------------------------------------------------------
void sub_405ACA()
{
  InitializeSListHead(&ListHead);
}

//----- (00E35AD6) --------------------------------------------------------
char sub_405AD6()
{
  return 1;
}

//----- (00E35AFA) --------------------------------------------------------
void *sub_405AFA()
{
  return &unk_E39100;
}

//----- (00E35B1D) --------------------------------------------------------
BOOL sub_405B1D()
{
  return dword_E39010 == 0;
}
// E39010: using guessed type int dword_409010;
```

```
//----- (00E35B29) --------------------------------------------------------
void *sub_405B29()
{
  return &unk_E394B0;
}

//----- (00E35B2F) --------------------------------------------------------
void *sub_405B2F()
{
  return &unk_E394AC;
}

//----- (00E35B35) --------------------------------------------------------
void sub_405B35()
{
  ;
}
// E35B35: could not find valid save-restore pair for edi

//----- (00E35B61) --------------------------------------------------------
void __cdecl sub_405B61()
{
  ;
}
// E35B61: could not find valid save-restore pair for edi

//----- (00E363A3) --------------------------------------------------------
unsigned __int64 __usercall sub_E363A3@<edx:eax>(unsigned __int64 a1@<edx:eax>)
{
  int v1; // ecx
  bool v2; // cc
  char v3; // cl

  v1 = HIDWORD(a1) >> 20;
  HIDWORD(a1) = HIDWORD(a1) & 0xFFFFF | 0x100000;
  v2 = v1 < 1075;
  v3 = v1 - 51;
  if ( v2 )
    return a1 >> (-v3 & 0x1F);
  else
    return a1 << (v3 & 0x1F);
}

//----- (00E36612) --------------------------------------------------------
void __cdecl sub_406612()
{
  sub_E319DE(dword_E3944C);
}
// E3944C: using guessed type int dword_40944C[3];

//----- (00E36621) --------------------------------------------------------
void __cdecl sub_406621()
{
  sub_E319DE(dword_E39458);
```

```
}
// E39458: using guessed type int dword_409458[3];

//----- (00E36630) --------------------------------------------------------
void __cdecl sub_406630()
{
  sub_E319DE(dword_E394A0);
}
// E394A0: using guessed type int dword_4094A0[3];

//----- (00E3663F) --------------------------------------------------------
void __cdecl sub_40663F()
{
  sub_E319DE(dword_E39440);
}
// E39440: using guessed type int dword_409440[3];

//----- (00E3664E) --------------------------------------------------------
void __cdecl sub_40664E()
{
  sub_E319DE(dword_E39464);
}
// E39464: using guessed type int dword_409464[3];

//----- (00E3665D) --------------------------------------------------------
void __cdecl sub_40665D()
{
  sub_E319DE(dword_E39494);
}
// E39494: using guessed type int dword_409494[3];

//----- (00E3666C) --------------------------------------------------------
void __cdecl sub_40666C()
{
  sub_E319DE(dword_E39434);
}
// E39434: using guessed type int dword_409434[3];

//----- (00E3667B) --------------------------------------------------------
void __cdecl sub_40667B()
{
  sub_E319DE(dword_E39488);
}
// E39488: using guessed type int dword_409488[3];

//----- (00E3668A) --------------------------------------------------------
void __cdecl sub_40668A()
{
  gEncode((void **)dword_E39470);
}
// E39470: using guessed type unsigned int dword_409470[6];

//----- (00E36699) --------------------------------------------------------
void __cdecl sub_406699()
{
```

```
  std::_Fac_tidy_reg_t::~_Fac_tidy_reg_t((std::_Fac_tidy_reg_t *)&unk_E390BC);
}

// nfuncs=356 queued=265 decompiled=265 lumina nreq=0 worse=0 better=0
#error "There were 1 decompilation failure(s) on 265 function(s)"
```