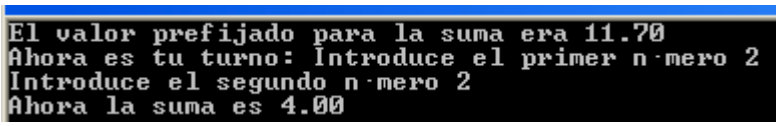
JUAN EL VIEJO
CRACKSLATINOS

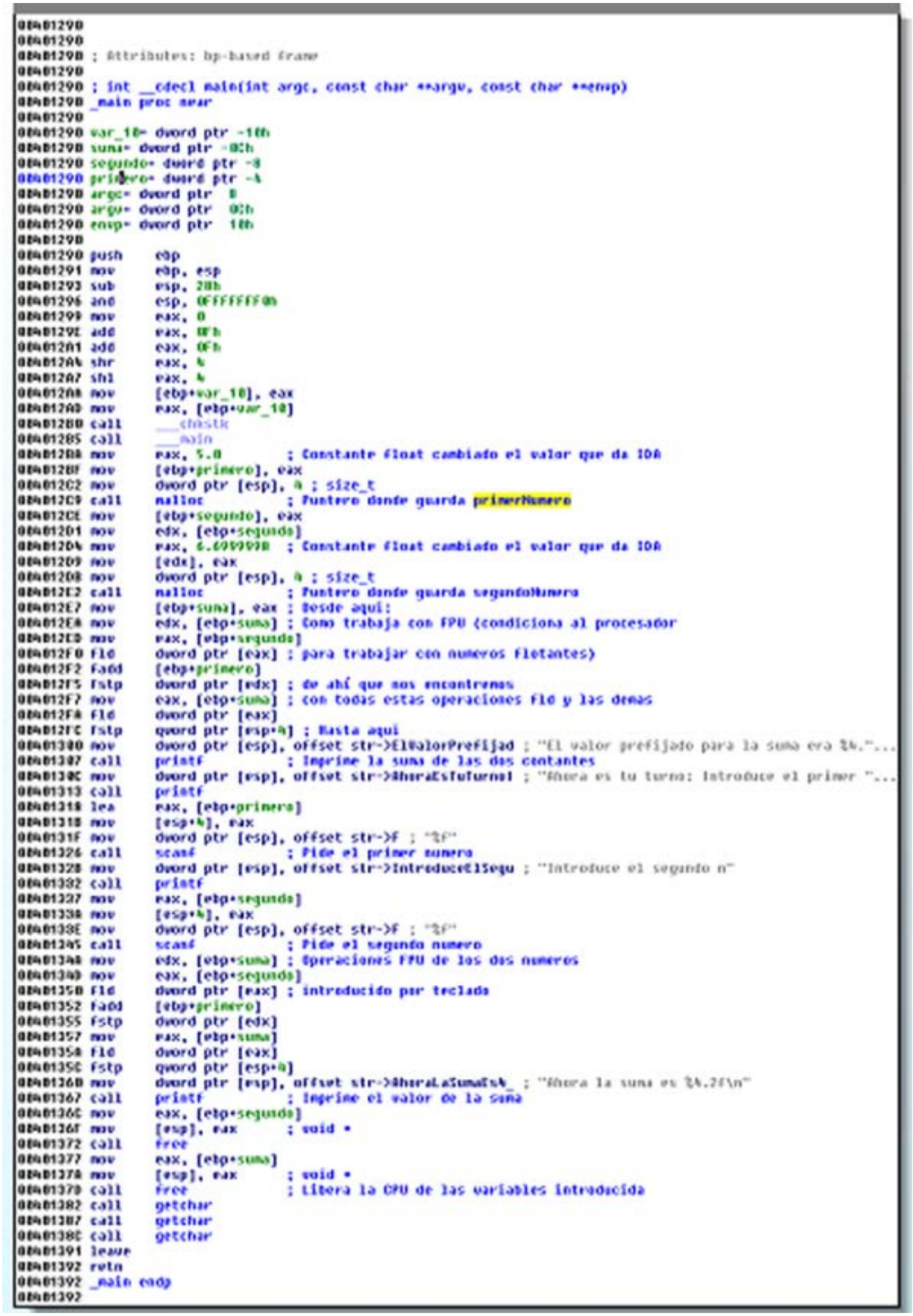
Programa	Ejemplo punteros.exe
Herramientas	IDA-Hex-Rays-DevC++
Protección	
Objetivo	Reversear

Ejecutamos el exe para ver lo que hace, vemos que de salida suma dos números flotantes y después suma dos números introducidos por teclado y el resultado lo pasa a flotante.



Desensamblado en IDA, mirando la Function name vemos que todo está en main.

Aquí resumo a mí entender lo que hace el programa.



```
00401290
00401290
00401290 ; Attributes: bp-based frame
00401290
00401290 ; int __cdecl main(int argc, const char **argv, const char **envp)
00401290 _main proc near
00401290
00401290 var_10= dword ptr -10h
00401290 suma= dword ptr -0Ch
00401290 segundo= dword ptr -8
00401290 primero= dword ptr -4
00401290 argc= dword ptr  8
00401290 argv= dword ptr  0Ch
00401290 envp= dword ptr  10h
00401290
00401290 push    ebp
00401291 mov     ebp, esp
00401293 sub     esp, 28h
00401296 and     esp, 0FFFFFFF0h
00401299 mov     eax, 0
0040129E add     eax, 0Fh
004012A1 add     eax, 0Fh
004012A4 shr     eax, 4
004012A7 shl     eax, 4
004012A8 mov     [ebp+var_10], eax
004012AB mov     eax, [ebp+var_10]
004012B0 call    __chkstk
004012B5 call    __main
004012BA mov     eax, 5.0        ; Constante float cambiado el valor que da IDA
004012BF mov     [ebp+primero], eax
004012C2 mov     dword ptr [esp], 4 ; size_t
004012C9 call    malloc          ; Puntero donde guarda primerNumero
004012CE mov     [ebp+segundo], eax
004012D1 mov     edx, [ebp+segundo]
004012D4 mov     eax, 6.099999E  ; Constante float cambiado el valor que da IDA
004012D9 mov     [edx], eax
004012DB mov     dword ptr [esp], 4 ; size_t
004012E2 call    malloc          ; Puntero donde guarda segundoNumero
004012E7 mov     [ebp+suma], eax ; Desde aqui:
004012EA mov     edx, [ebp+suma] ; Como trabaja con FPU (condiciona al procesador
004012ED mov     eax, [ebp+segundo]
004012F0 fld     dword ptr [eax] ; para trabajar con numeros fletantes)
004012F2 fadd    [ebp+primero]
004012F5 fstp    dword ptr [edx] ; de ahí que nos encontremos
004012F7 mov     eax, [ebp+suma] ; con todas estas operaciones fld y las demas
004012FA fld     dword ptr [eax]
004012FC fstp    qword ptr [esp+4] ; Hasta aqui
00401300 mov     dword ptr [esp], offset str->ElValorPrefijad ; "El valor prefijado para la suma era %4."...
00401307 call    printf          ; Imprime la suma de las dos constantes
0040130C mov     dword ptr [esp], offset str->AhoraEsTuTurno1 ; "Ahora es tu turno: Introduce el primer "...
00401313 call    printf
00401318 lea     eax, [ebp+primero]
0040131B mov     [esp+4], eax
0040131F mov     dword ptr [esp], offset str->f ; "%f"
00401326 call    scanf           ; Pide el primer numero
00401328 mov     dword ptr [esp], offset str->IntroduceElSegu ; "Introduce el segundo n"
00401332 call    printf
00401337 mov     eax, [ebp+segundo]
0040133A mov     [esp+4], eax
0040133E mov     dword ptr [esp], offset str->f ; "%f"
00401345 call    scanf           ; Pide el segundo numero
00401348 mov     edx, [ebp+suma] ; Operaciones FPU de los dos numeros
0040134B mov     eax, [ebp+segundo]
0040135B fld     dword ptr [eax] ; introducido por teclado
00401352 fadd    [ebp+primero]
00401355 fstp    dword ptr [edx]
00401357 mov     eax, [ebp+suma]
0040135A fld     dword ptr [eax]
0040135C fstp    qword ptr [esp+4]
00401360 mov     dword ptr [esp], offset str->AhoraLaSumaEs%_ ; "Ahora la suma es %4.2f\n"
00401367 call    printf          ; Imprime el valor de la suma
0040136C mov     eax, [ebp+segundo]
0040136F mov     [esp], eax      ; void *
00401372 call    free
00401377 mov     eax, [ebp+suma]
0040137A mov     [esp], eax      ; void *
0040137D call    free            ; Libera la CPU de las variables introducida
00401382 call    getchar
00401387 call    getchar
0040138C call    getchar
00401391 leave
00401392 retn
00401392 _main endp
00401392
```

Esta es la ayuda del Hex-Rays, siempre compruebo con el plugin si voy por buen camino.

Ahora me queda programarlo en C.
Le he añadido unos toques para aclarar donde están los punteros (*).

```
void *v4; // esp@1
  void *v5; // eax@1
  void *v6; // ST20_4@1
  void *v7; // eax@1
  void *v8; // ST1C_4@1
  float primero; // [sp+24h] [bp-4h]@1

  v4 = alloca(16);
  __main();
  LODWORD(primero) = 1084227584;  // 5.0
  v5 = malloc(4u);
  v6 = v5;
  *(_DWORD *)v5 = 1087792742;  // *segundo 6.7
  v7 = malloc(4u);
  v8 = v7;
  *(float *)v7 = *(float *)v6 + primero;
  printf("El valor prefijado para la suma era %4.2f\n", *(float *)v7); // *(float
*)v7=*suma
  printf("Ahora es tu turno: Introduce el primer n");
  scanf("%f", &primero);
  printf("Introduce el segundo n");
  scanf("%f", v6); // *segundo
  *(float *)v8 = *(float *)v6 + primero;  // *suma = *segundo+primero
  printf("Ahora la suma es %4.2f\n", *(float *)v8);  //*(float *)v8= *suma
  free(v6); // libera segundo
  free(v8); // libera suma
  getchar();
  getchar();
  return getchar();
```

Mando el reverseado y también el código.

Hasta aquí llegué, un saludo a toda la lista. Juan