28.12.2023

Курс:

Практическая работа к уроку № Lesson_5

--
Реверс-инжиниринг с помощью OllyDbg

Задание:

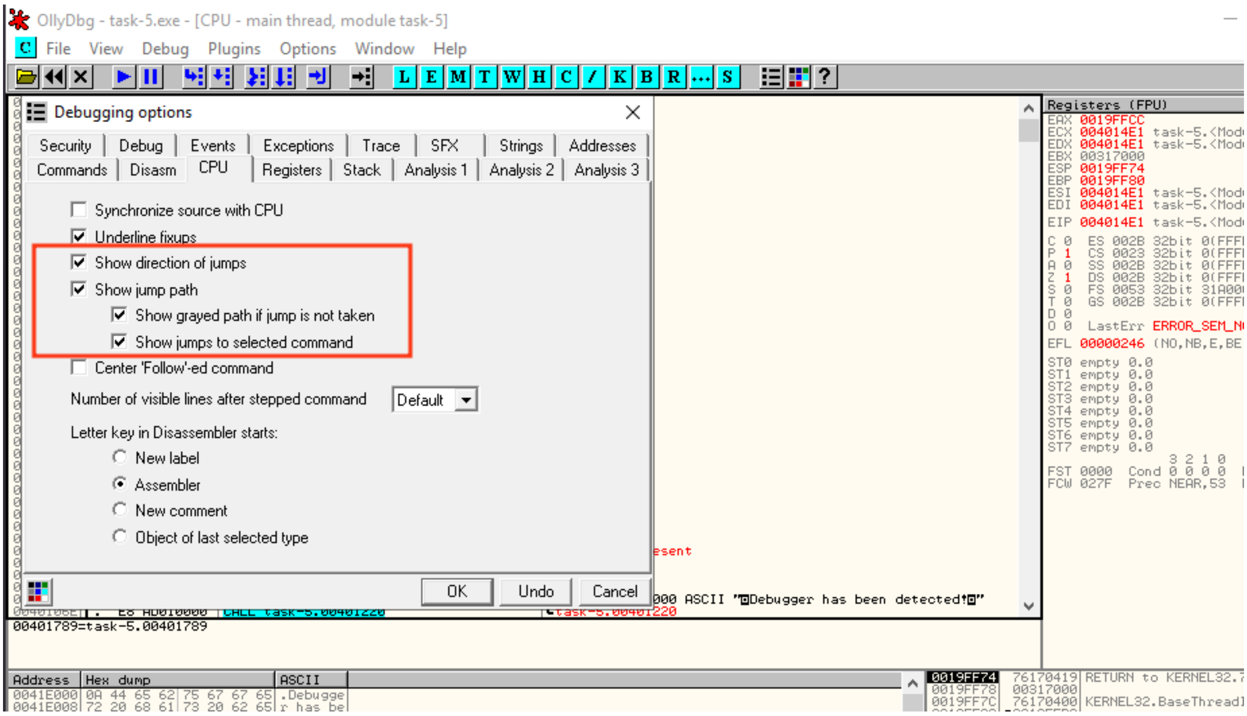
Дана программа task-3. Необходимо выполнить реверс-инжиниринг программы и написать ее псевдокод.

Пример запуска программы:
task-3.exe

Insert name: Sergey
License key: TIfKsMhOfQzShUiW

Настройки ***OllyDbg***:

- Подсветка: Appearance -> Highlighting -> Jumps n' calls (правая кнопка мыши)
- Debugging options -> CPU (указатели)


OllyDbg - task-5.exe - [CPU - main thread, module task-5]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
Security Debug Events Exceptions Trace SFX Strings Addresses
Commands Disasm CPU Registers Stack Analysis 1 Analysis 2 Analysis 3
OK Undo Cancel

Debugging options

- ☐ Synchronize source with CPU
- ☑ Underline fixups
- ☑ Show direction of jumps
- ☑ Show jump path
  - ☑ Show grayed path if jump is not taken
  - ☑ Show jumps to selected command
- ☐ Center 'Follow'-ed command

Number of visible lines after stepped command: Default

Letter key in Disassembler starts:

- ○ New label
- ● Assembler
- ○ New comment
- ○ Object of last selected type

```
esent
000 ASCII "Debugger has been detected!"
00401220
00401789=task-5.00401789
```

```
Registers (FPU)
EAX 0019FFCC
ECX 004014E1 task-5.<Mod
EDX 004014E1 task-5.<Mod
EBX 00317000
ESP 0019FF74
EBP 0019FF80
ESI 004014E1 task-5.<Mod
EDI 004014E1 task-5.<Mod

EIP 004014E1 task-5.<Mod

C 0  ES 002B 32bit 0(FFF
P 1  CS 0023 32bit 0(FFF
A 0  SS 002B 32bit 0(FFF
Z 1  DS 002B 32bit 0(FFF
S 0  FS 0053 32bit 31A00
T 0  GS 002B 32bit 0(FFF
D 0
O 0  LastErr ERROR_SEM_N

EFL 00000246 (NO,NB,E,BE

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0
FST 0000  Cond 0 0 0 0
FCW 027F  Prec NEAR,53
```

```
Address  Hex dump                 ASCII
0041E000 0A 44 65 62 75 67 67 65 .Debugge
0041E008 72 20 68 61 73 20 62 65 r has be
```

```
0019FF74  76170419 RETURN to KERNEL32.
0019FF78  00317000
0019FF7C  76170400 KERNEL32.BaseThread
```

Commands:

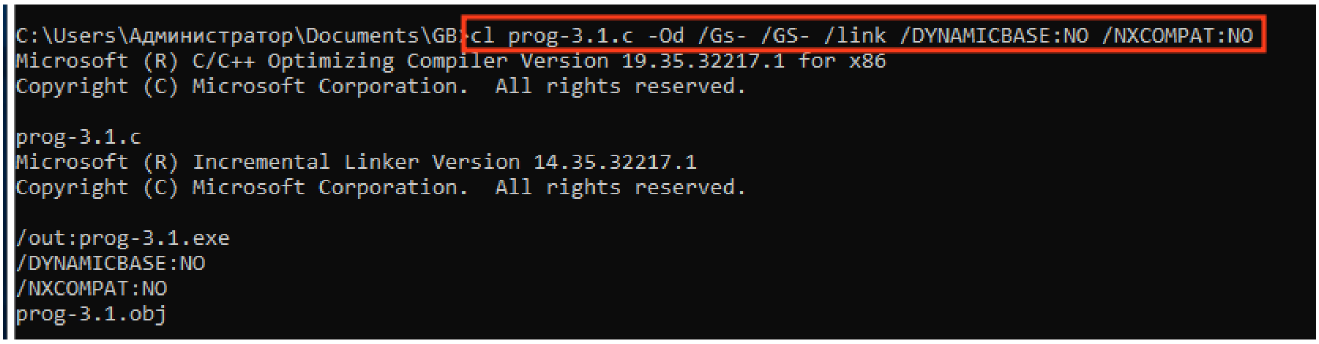
- F2 точка останова
- F8 одновременное выполнение вызовов функций
- F9 продолжает программу заполнения
- Ctrl + F2 перезапустим программу
- Ctrl + G Enter expression to follow in Dump

Отладочные символы:

- Debug -> Select path for symbols
  копируем prog-3.3.pdb в папку

```
// Исходный код программы prog-3.1.c

#include <windows.h>
#pragma comment(lib,"user32.lib")

int main(int argc, char* argv[]) {
int tmp = 2;
char title[] = "Prog-3.1";

goto metka;
        MessageBox(NULL, "Always Skipped", title, 0);
metka:
if (TRUE) { MessageBox(NULL, "Always TRUE", title, 0); }
if (FALSE) { MessageBox(NULL, "Always FALSE", title, 0); }
if (tmp < 5) { MessageBox(NULL, "tmp < 5", title, 0); }
if (tmp > 5) {
        MessageBox(NULL, "tmp > 5", title, 0);
} else {
        MessageBox(NULL, "tmp <= 5", title, 0);
}
if (tmp == 2) { MessageBox(NULL, "tmp == 2", title, 0); }
switch (tmp)
{
        case 1:
                MessageBox(NULL, "Case: 1", title, 0);
                break;
        case 2:
                MessageBox(NULL, "Case: 2", title, 0);
                break;
        default:
                MessageBox(NULL, "Case: default", title, 0);
                break;
}
return 0;
}
```

```
Компиляция:
> cl prog-3.1.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
Запуск:
> prog-3.1.exe
```



```
C:\Users\Администратор\Documents\GB>cl prog-3.1.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
Microsoft (R) C/C++ Optimizing Compiler Version 19.35.32217.1 for x86
Copyright (C) Microsoft Corporation.  All rights reserved.

prog-3.1.c
Microsoft (R) Incremental Linker Version 14.35.32217.1
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:prog-3.1.exe
/DYNAMICBASE:NO
/NXCOMPAT:NO
prog-3.1.obj
```

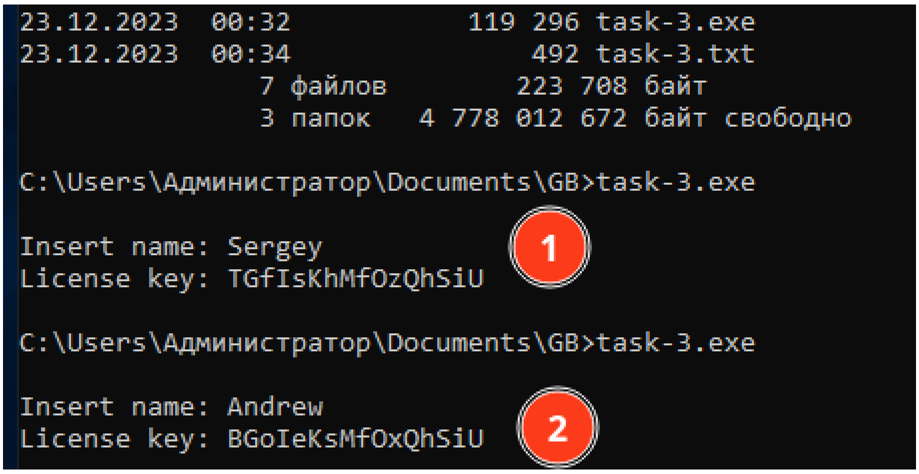
```c
//prog-3.2.c

#include <stdio.h>
#pragma comment(lib,"user32.lib")
int sum_ascii(char *);
int main(int argc, char* argv[]) {
int result = 0;
char title[] = "Prog-3.3";
if (argc < 2) {
printf("\nUsage: prog-3.3.exe <license key>\n");
return 0;
}
result = sum_ascii(argv[1]);
if (result == 1000) {
printf("\nCongrats! Key is correct!\n");
} else {
printf("\nOops! Key is incorrect!\n");
}
return 0;
}
int sum_ascii(char *str) {
int sum = 0;
int len = strlen(str);
for(int i = 0; i < len; i++) {
sum += str[i];
}
return sum;
}
```

```
Компиляция:
> cl prog-3.2.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
```

1. Запускаем в CMD *task-3.exe*

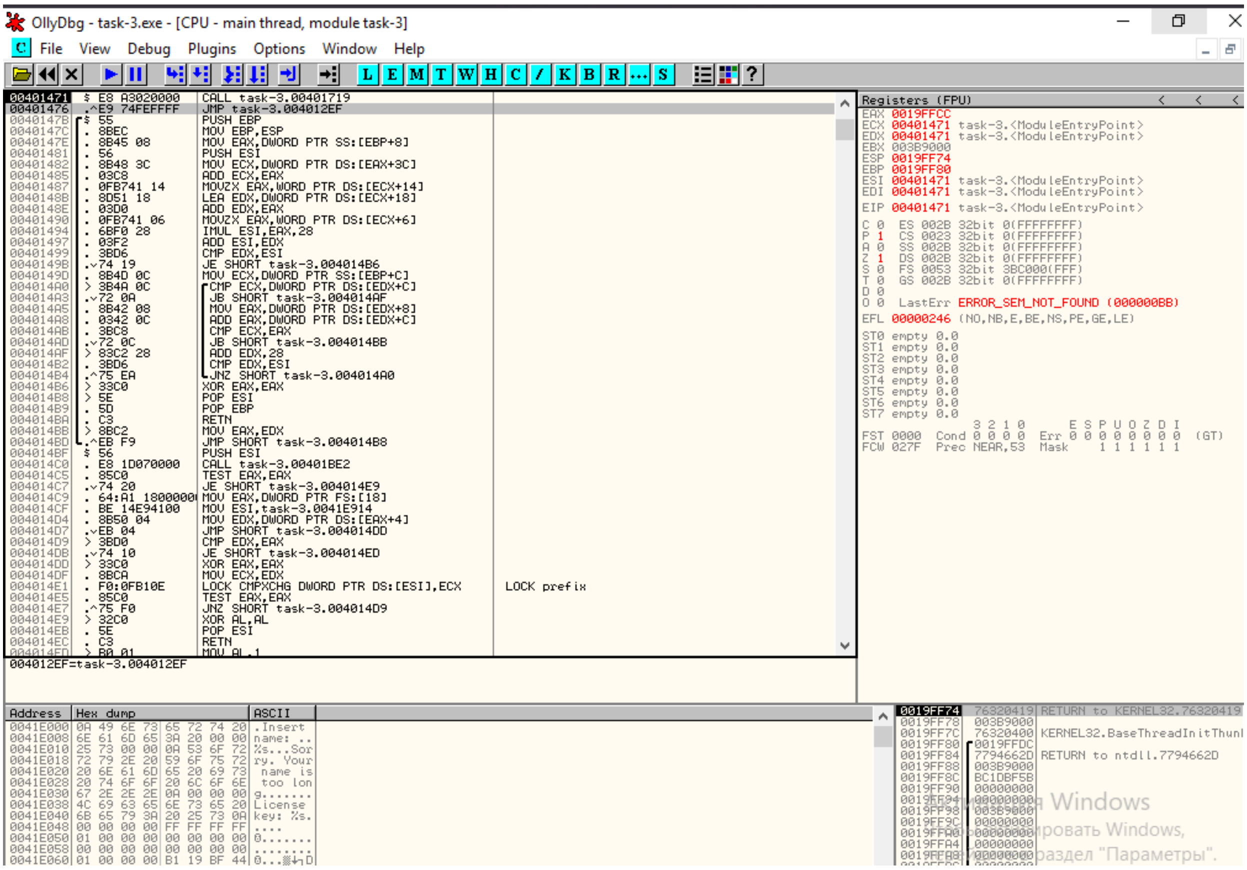
Insert name: Sergey
License key: TIfKsMhOfQzShUiW

Пробуем ввести другое имя (2)



```
23.12.2023  00:32           119 296 task-3.exe
23.12.2023  00:34               492 task-3.txt
               7 файлов        223 708 байт
               3 папок   4 778 012 672 байт свободно

C:\Users\Администратор\Documents\GB>task-3.exe

Insert name: Sergey
License key: TGfIsKhMfOzQhSiU

C:\Users\Администратор\Documents\GB>task-3.exe

Insert name: Andrew
License key: BGoIeKsMfOxQhSiU
```

2. Запускаем в OllyDbg *task-3.exe*



OllyDbg - task-3.exe - [CPU - main thread, module task-3]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

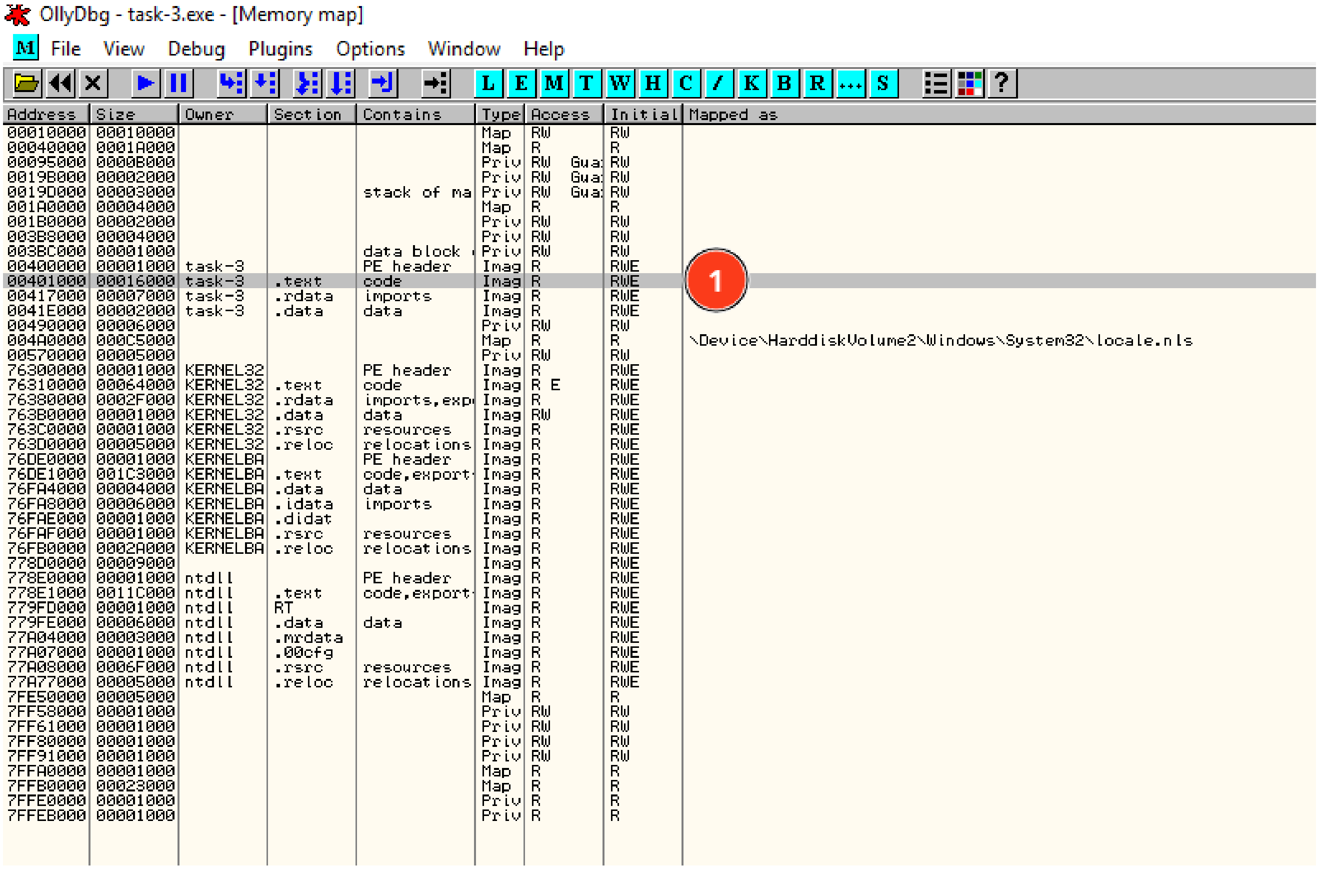
```
00401471  $ E8 A3020000    CALL task-3.00401719
00401476  .^E9 74FEFFFF    JMP task-3.004012EF
0040147B  $ 55             PUSH EBP
0040147C  . 8BEC           MOV EBP,ESP
0040147E  . 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
00401481  . 56             PUSH ESI
00401482  . 8B48 3C        MOV ECX,DWORD PTR DS:[EAX+3C]
00401485  . 03C8           ADD ECX,EAX
00401487  . 0FB741 14      MOVZX EAX,WORD PTR DS:[ECX+14]
0040148B  . 8D51 18        LEA EDX,DWORD PTR DS:[ECX+18]
0040148E  . 03D0           ADD EDX,EAX
00401490  . 0FB741 06      MOVZX EAX,WORD PTR DS:[ECX+6]
00401494  . 6BF0 28        IMUL ESI,EAX,28
00401497  . 03F2           ADD ESI,EDX
00401499  . 3BD6           CMP EDX,ESI
0040149B  .v74 19          JE SHORT task-3.004014B6
0040149D  . 8B4D 0C        MOV ECX,DWORD PTR SS:[EBP+C]
004014A0  > 3B4A 0C        CMP ECX,DWORD PTR DS:[EDX+C]
004014A3  .v72 0A           JB SHORT task-3.004014AF
004014A5  . 8B42 08         MOV EAX,DWORD PTR DS:[EDX+8]
004014A8  . 0342 0C         ADD EAX,DWORD PTR DS:[EDX+C]
004014AB  . 3BC8            CMP ECX,EAX
004014AD  .v72 0C           JB SHORT task-3.004014BB
004014AF  > 83C2 28         ADD EDX,28
004014B2  . 3BD6            CMP EDX,ESI
004014B4  .^75 EA           JNZ SHORT task-3.004014A0
004014B6  > 33C0           XOR EAX,EAX
004014B8  > 5E             POP ESI
004014B9  . 5D             POP EBP
004014BA  . C3             RETN
004014BB  > 8BC2           MOV EAX,EDX
004014BD  .^EB F9          JMP SHORT task-3.004014B8
004014BF  $ 56             PUSH ESI
004014C0  . E8 1D070000    CALL task-3.00401BE2
004014C5  . 85C0           TEST EAX,EAX
004014C7  .v74 20          JE SHORT task-3.004014E9
004014C9  . 64:A1 18000000 MOV EAX,DWORD PTR FS:[18]
004014CF  . BE 14E94100    MOV ESI,task-3.0041E914
004014D4  . 8B50 04        MOV EDX,DWORD PTR DS:[EAX+4]
004014D7  .vEB 04          JMP SHORT task-3.004014DD
004014D9  > 3BD0           CMP EDX,EAX
004014DB  .v74 10          JE SHORT task-3.004014ED
004014DD  > 33C0           XOR EAX,EAX
004014DF  . 8BCA           MOV ECX,EDX
004014E1  . F0:0FB10E      LOCK CMPXCHG DWORD PTR DS:[ESI],ECX      LOCK prefix
004014E5  . 85C0           TEST EAX,EAX
004014E7  .^75 F0          JNZ SHORT task-3.004014D9
004014E9  > 32C0           XOR AL,AL
004014EB  . 5E             POP ESI
004014EC  . C3             RETN
004014ED  > B0 01          MOV AL,1
004012EF=task-3.004012EF
```

```
Registers (FPU)
EAX 0019FFCC
ECX 00401471 task-3.<ModuleEntryPoint>
EDX 00401471 task-3.<ModuleEntryPoint>
EBX 003B9000
ESP 0019FF74
EBP 0019FF80
ESI 00401471 task-3.<ModuleEntryPoint>
EDI 00401471 task-3.<ModuleEntryPoint>

EIP 00401471 task-3.<ModuleEntryPoint>

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3BC000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SEM_NOT_FOUND (000000BB)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
                 3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address   Hex dump                 ASCII
0041E000  0A 49 6E 73 65 72 74 20  .Insert
0041E008  6E 61 6D 65 3A 20 00 00  name: ..
0041E010  25 73 00 00 0A 53 6F 72  %s...Sor
0041E018  72 79 2E 20 59 6F 75 72  ry. Your
0041E020  20 6E 61 6D 65 20 69 73   name is
0041E028  20 74 6F 6F 20 6C 6F 6E   too lon
0041E030  67 2E 2E 2E 0A 00 00 00  g.......
0041E038  4C 69 63 65 6E 73 65 20  License
0041E040  6B 65 79 3A 20 25 73 0A  key: %s.
0041E048  00 00 00 00 FF FF FF FF  ....
0041E050  01 00 00 00 00 00 00 00  0.......
0041E058  00 00 00 00 00 00 00 00  ........
0041E060  01 00 00 00 B1 19 BF 44  0...▒↓┐D
```

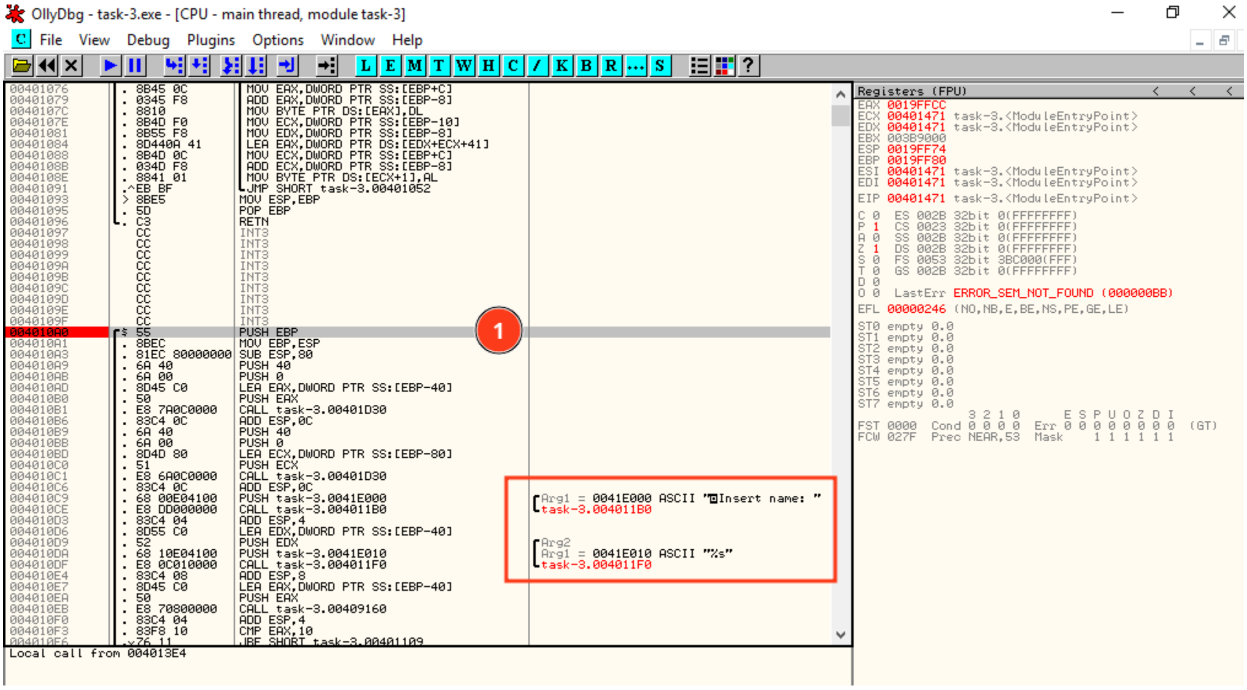
```
0019FF74  76320419  RETURN to KERNEL32.76320419
0019FF78  003B9000
0019FF7C  76320400  KERNEL32.BaseThreadInitThunk
0019FF80  0019FFDC
0019FF84  7794662D  RETURN to ntdll.7794662D
0019FF88  003B9000
0019FF8C  BC1DBF5B
0019FF90  00000000
0019FF94  00000000
0019FF98  003B9000
0019FF9C  00000000
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
```

Далее переходим в раздел *(M)emory*


OllyDbg - task-3.exe - [Memory map]
M File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
1

```
Address  Size     Owner    Section  Contains     Type Access    Initial Mapped as
00010000 00010000                                Map  RW        RW
00040000 0001A000                                Map  R         R
00095000 0000B000                                Priv RW   Gua: RW
0019B000 00002000                                Priv RW   Gua: RW
0019D000 00003000                   stack of ma  Priv RW   Gua: RW
001A0000 00004000                                Map  R         R
001B0000 00002000                                Priv RW        RW
003B8000 00004000                                Priv RW        RW
003BC000 00001000                   data block ( Priv RW        RW
00400000 00001000 task-3            PE header    Imag R         RWE
00401000 00016000 task-3   .text    code         Imag R         RWE
00417000 00007000 task-3   .rdata   imports      Imag R         RWE
0041E000 00002000 task-3   .data    data         Imag R         RWE
00490000 00006000                                Priv RW        RW
004A0000 000C5000                                Map  R         R       \Device\HarddiskVolume2\Windows\System32\locale.nls
00570000 00005000                                Priv RW        RW
76300000 00001000 KERNEL32          PE header    Imag R         RWE
76310000 00064000 KERNEL32 .text    code         Imag R E       RWE
76380000 0002F000 KERNEL32 .rdata   imports,exp( Imag R         RWE
763B0000 00001000 KERNEL32 .data    data         Imag RW        RWE
763C0000 00001000 KERNEL32 .rsrc    resources    Imag R         RWE
763D0000 00005000 KERNEL32 .reloc   relocations  Imag R         RWE
76DE0000 00001000 KERNELBA          PE header    Imag R         RWE
76DE1000 001C3000 KERNELBA .text    code,export: Imag R         RWE
76FA4000 00004000 KERNELBA .data    data         Imag R         RWE
76FA8000 00006000 KERNELBA .idata   imports      Imag R         RWE
76FAE000 00001000 KERNELBA .didat                Imag R         RWE
76FAF000 00001000 KERNELBA .rsrc    resources    Imag R         RWE
76FB0000 0002A000 KERNELBA .reloc   relocations  Imag R         RWE
778D0000 00009000                                Imag R         RWE
778E0000 00001000 ntdll             PE header    Imag R         RWE
778E1000 0011C000 ntdll    .text    code,export: Imag R         RWE
779FD000 00001000 ntdll    RT                    Imag R         RWE
779FE000 00006000 ntdll    .data    data         Imag R         RWE
77A04000 00003000 ntdll    .mrdata               Imag R         RWE
77A07000 00001000 ntdll    .00cfg                Imag R         RWE
77A08000 0006F000 ntdll    .rsrc    resources    Imag R         RWE
77A77000 00005000 ntdll    .reloc   relocations  Imag R         RWE
7FE50000 00005000                                Map  R         R
7FF58000 00001000                                Priv RW        RW
7FF61000 00001000                                Priv RW        RW
7FF80000 00001000                                Priv RW        RW
7FF91000 00001000                                Priv RW        RW
7FFA0000 00001000                                Map  R         R
7FFB0000 00023000                                Map  R         R
7FFE0000 00001000                                Priv R         R
7FFEB000 00001000                                Priv R         R
```

4. Сердце кода находится по адресу 00401000 и переходим по нему *(C)*

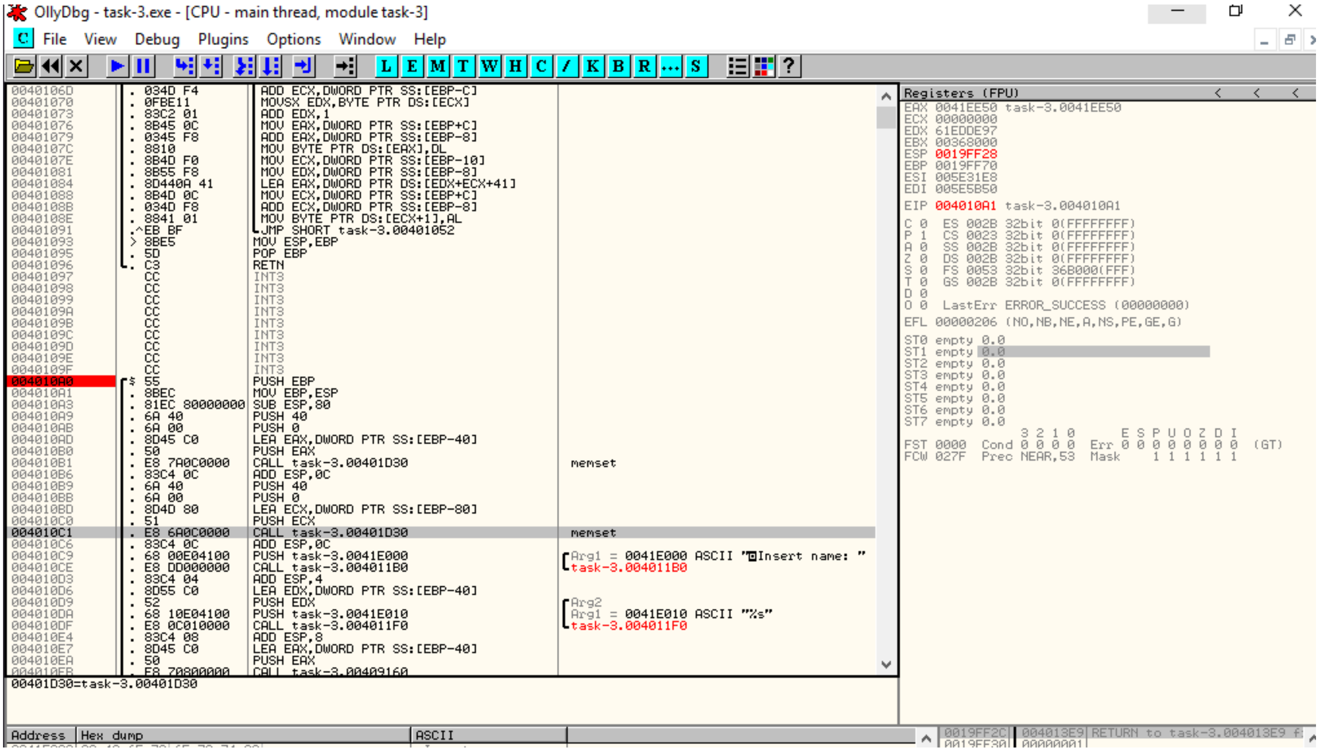

OllyDbg - task-3.exe - [CPU - main thread, module task-3]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
1

```
00401076  . 8B45 0C        MOV EAX,DWORD PTR SS:[EBP+C]
00401079  . 0345 F8        ADD EAX,DWORD PTR SS:[EBP-8]
0040107C  . 8810           MOV BYTE PTR DS:[EAX],DL
0040107E  . 8B4D F0        MOV ECX,DWORD PTR SS:[EBP-10]
00401081  . 8B55 F8        MOV EDX,DWORD PTR SS:[EBP-8]
00401084  . 8D440A 41      LEA EAX,DWORD PTR DS:[EDX+ECX+41]
00401088  . 8B4D 0C        MOV ECX,DWORD PTR SS:[EBP+C]
0040108B  . 034D F8        ADD ECX,DWORD PTR SS:[EBP-8]
0040108E  . 8841 01        MOV BYTE PTR DS:[ECX+1],AL
00401091  .^EB BF          JMP SHORT task-3.00401052
00401093  > 8BE5           MOV ESP,EBP
00401095  . 5D             POP EBP
00401096  . C3             RETN
00401097    CC             INT3
00401098    CC             INT3
00401099    CC             INT3
0040109A    CC             INT3
0040109B    CC             INT3
0040109C    CC             INT3
0040109D    CC             INT3
0040109E    CC             INT3
0040109F    CC             INT3
004010A0  $ 55             PUSH EBP
004010A1  . 8BEC           MOV EBP,ESP
004010A3  . 81EC 80000000  SUB ESP,80
004010A9  . 6A 40          PUSH 40
004010AB  . 6A 00          PUSH 0
004010AD  . 8D45 C0        LEA EAX,DWORD PTR SS:[EBP-40]
004010B0  . 50             PUSH EAX
004010B1  . E8 7A0C0000    CALL task-3.00401D30
004010B6  . 83C4 0C        ADD ESP,0C
004010B9  . 6A 40          PUSH 40
004010BB  . 6A 00          PUSH 0
004010BD  . 8D4D 80        LEA ECX,DWORD PTR SS:[EBP-80]
004010C0  . 51             PUSH ECX
004010C1  . E8 6A0C0000    CALL task-3.00401D30
004010C6  . 83C4 0C        ADD ESP,0C
004010C9  . 68 00E04100    PUSH task-3.0041E000                  [Arg1 = 0041E000 ASCII "◘Insert name: "
004010CE  . E8 DD000000    CALL task-3.004011B0                  [task-3.004011B0
004010D3  . 83C4 04        ADD ESP,4
004010D6  . 8D55 C0        LEA EDX,DWORD PTR SS:[EBP-40]
004010D9  . 52             PUSH EDX                              [Arg2
004010DA  . 68 10E04100    PUSH task-3.0041E010                  |Arg1 = 0041E010 ASCII "%s"
004010DF  . E8 0C010000    CALL task-3.004011F0                  [task-3.004011F0
004010E4  . 83C4 08        ADD ESP,8
004010E7  . 8D45 C0        LEA EAX,DWORD PTR SS:[EBP-40]
004010EA  . 50             PUSH EAX
004010EB  . E8 70800000    CALL task-3.00409160
004010F0  . 83C4 04        ADD ESP,4
004010F3  . 83F8 10        CMP EAX,10
004010F6  .v76 11          JBE SHORT task-3.00401109
Local call from 004013E4
```

```
Registers (FPU)
EAX 0019FFCC
ECX 00401471 task-3.<ModuleEntryPoint>
EDX 00401471 task-3.<ModuleEntryPoint>
EBX 003B9000
ESP 0019FF74
EBP 0019FF80
ESI 00401471 task-3.<ModuleEntryPoint>
EDI 00401471 task-3.<ModuleEntryPoint>

EIP 00401471 task-3.<ModuleEntryPoint>

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3BC000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SEM_NOT_FOUND (000000BB)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

Находим блок и устанавливаем точку остановки (break point), где начинается функция $PUSH$ и блок, как в нашем случае, с $Arg1$... $Arg2$
Отмеченный блок напоминает функцию $main$

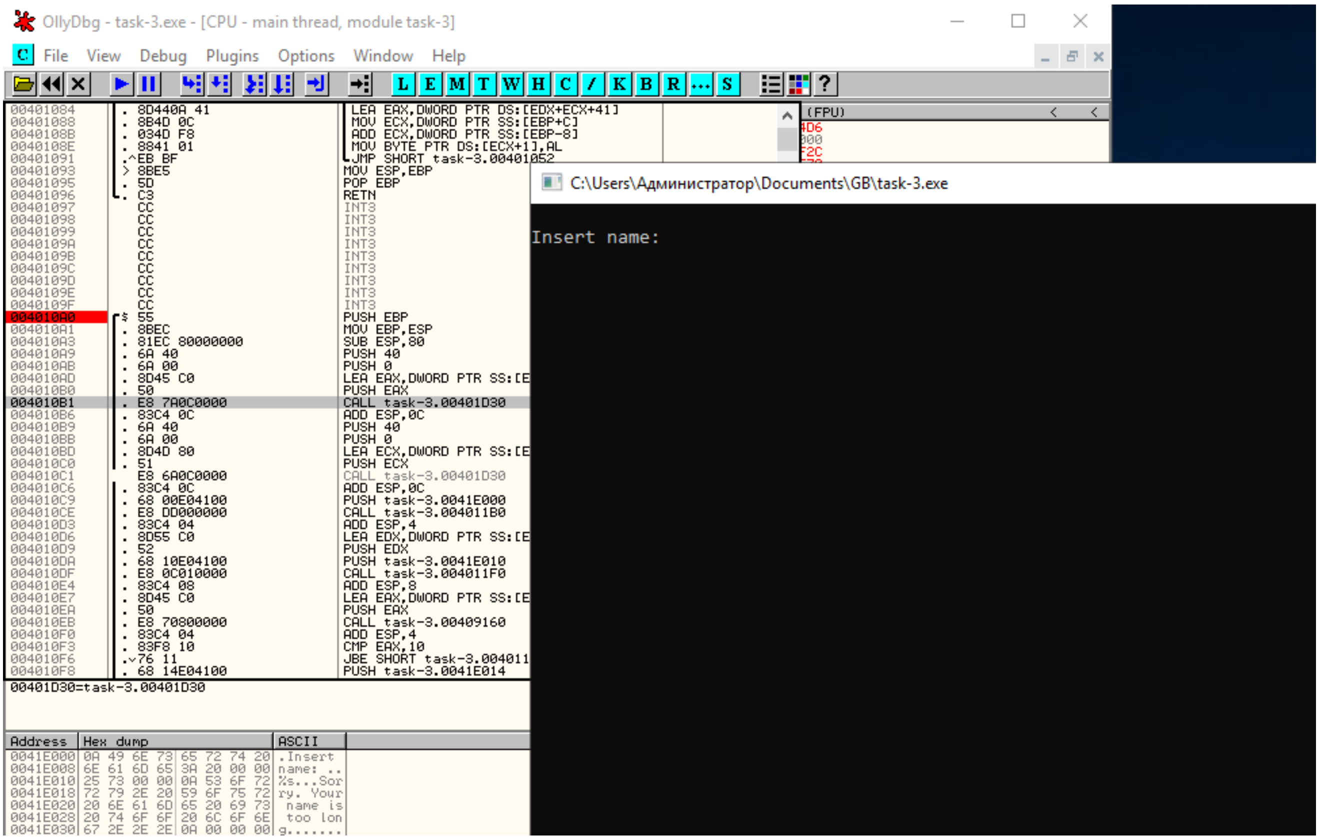
5. Запускаем программу
6. Начинаем изучать код для подготовки псевдокода
   функции похожи на $memset$, комментируем в коде...


OllyDbg - task-3.exe - [CPU - main thread, module task-3]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Registers (FPU)
Address Hex dump ASCII

```
0040106D  . 034D F4         ADD ECX,DWORD PTR SS:[EBP-C]
00401070  . 0FBE11          MOVSX EDX,BYTE PTR DS:[ECX]
00401073  . 83C2 01         ADD EDX,1
00401076  . 8B45 0C         MOV EAX,DWORD PTR SS:[EBP+C]
00401079  . 0345 F8         ADD EAX,DWORD PTR SS:[EBP-8]
0040107C  . 8810            MOV BYTE PTR DS:[EAX],DL
0040107E  . 8B4D F0         MOV ECX,DWORD PTR SS:[EBP-10]
00401081  . 8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
00401084  . 8D440A 41       LEA EAX,DWORD PTR DS:[EDX+ECX+41]
00401088  . 8B4D 0C         MOV ECX,DWORD PTR SS:[EBP+C]
0040108B  . 034D F8         ADD ECX,DWORD PTR SS:[EBP-8]
0040108E  . 8841 01         MOV BYTE PTR DS:[ECX+1],AL
00401091  .^EB BF           JMP SHORT task-3.00401052
00401093  > 8BE5            MOV ESP,EBP
00401095  . 5D              POP EBP
00401096  . C3              RETN
00401097    CC              INT3
00401098    CC              INT3
00401099    CC              INT3
0040109A    CC              INT3
0040109B    CC              INT3
0040109C    CC              INT3
0040109D    CC              INT3
0040109E    CC              INT3
0040109F    CC              INT3
004010A0  $ 55              PUSH EBP
004010A1  . 8BEC            MOV EBP,ESP
004010A3  . 81EC 80000000   SUB ESP,80
004010A9  . 6A 40           PUSH 40
004010AB  . 6A 00           PUSH 0
004010AD  . 8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010B0  . 50              PUSH EAX
004010B1  . E8 7A0C0000     CALL task-3.00401D30                 memset
004010B6  . 83C4 0C         ADD ESP,0C
004010B9  . 6A 40           PUSH 40
004010BB  . 6A 00           PUSH 0
004010BD  . 8D4D 80         LEA ECX,DWORD PTR SS:[EBP-80]
004010C0  . 51              PUSH ECX
004010C1  . E8 6A0C0000     CALL task-3.00401D30                 memset
004010C6  . 83C4 0C         ADD ESP,0C
004010C9  . 68 00E04100     PUSH task-3.0041E000                 Arg1 = 0041E000 ASCII "◘Insert name: "
004010CE  . E8 DD000000     CALL task-3.004011B0                 task-3.004011B0
004010D3  . 83C4 04         ADD ESP,4
004010D6  . 8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
004010D9  . 52              PUSH EDX                             Arg2
004010DA  . 68 10E04100     PUSH task-3.0041E010                 Arg1 = 0041E010 ASCII "%s"
004010DF  . E8 0C010000     CALL task-3.004011F0                 task-3.004011F0
004010E4  . 83C4 08         ADD ESP,8
004010E7  . 8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010EA  . 50              PUSH EAX
004010EB  . E8 70800000     CALL task-3.00409160
00401D30=task-3.00401D30
```

```
EAX 0041EE50 task-3.0041EE50
ECX 00000000
EDX 61EDDE97
EBX 00368000
ESP 0019FF28
EBP 0019FF70
ESI 005E31E8
EDI 005E5B50
EIP 004010A1 task-3.004010A1
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 36B000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

0019FF2C  004013E9  RETURN to task-3.004013E9 f
0019FF30  00000001
```

Выполним вызов функции $0041E000$, видим, что она является $printf$

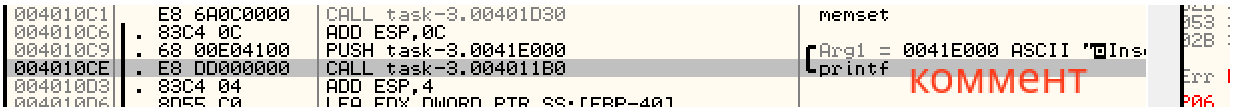

OllyDbg - task-3.exe - [CPU - main thread, module task-3]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
(FPU)
Address Hex dump ASCII

```
00401084  . 8D440A 41       LEA EAX,DWORD PTR DS:[EDX+ECX+41]
00401088  . 8B4D 0C         MOV ECX,DWORD PTR SS:[EBP+C]
0040108B  . 034D F8         ADD ECX,DWORD PTR SS:[EBP-8]
0040108E  . 8841 01         MOV BYTE PTR DS:[ECX+1],AL
00401091  .^EB BF           JMP SHORT task-3.00401052
00401093  > 8BE5            MOV ESP,EBP
00401095  . 5D              POP EBP
00401096  . C3              RETN
00401097    CC              INT3
00401098    CC              INT3
00401099    CC              INT3
0040109A    CC              INT3
0040109B    CC              INT3
0040109C    CC              INT3
0040109D    CC              INT3
0040109E    CC              INT3
0040109F    CC              INT3
004010A0  $ 55              PUSH EBP
004010A1  . 8BEC            MOV EBP,ESP
004010A3  . 81EC 80000000   SUB ESP,80
004010A9  . 6A 40           PUSH 40
004010AB  . 6A 00           PUSH 0
004010AD  . 8D45 C0         LEA EAX,DWORD PTR SS:[E
004010B0  . 50              PUSH EAX
004010B1  . E8 7A0C0000     CALL task-3.00401D30
004010B6  . 83C4 0C         ADD ESP,0C
004010B9  . 6A 40           PUSH 40
004010BB  . 6A 00           PUSH 0
004010BD  . 8D4D 80         LEA ECX,DWORD PTR SS:[E
004010C0  . 51              PUSH ECX
004010C1    E8 6A0C0000     CALL task-3.00401D30
004010C6  . 83C4 0C         ADD ESP,0C
004010C9  . 68 00E04100     PUSH task-3.0041E000
004010CE  . E8 DD000000     CALL task-3.004011B0
004010D3  . 83C4 04         ADD ESP,4
004010D6  . 8D55 C0         LEA EDX,DWORD PTR SS:[E
004010D9  . 52              PUSH EDX
004010DA  . 68 10E04100     PUSH task-3.0041E010
004010DF  . E8 0C010000     CALL task-3.004011F0
004010E4  . 83C4 08         ADD ESP,8
004010E7  . 8D45 C0         LEA EAX,DWORD PTR SS:[E
004010EA  . 50              PUSH EAX
004010EB  . E8 70800000     CALL task-3.00409160
004010F0  . 83C4 04         ADD ESP,4
004010F3  . 83F8 10         CMP EAX,10
004010F6  .v76 11           JBE SHORT task-3.004011
004010F8  . 68 14E04100     PUSH task-3.0041E014
00401D30=task-3.00401D30
```

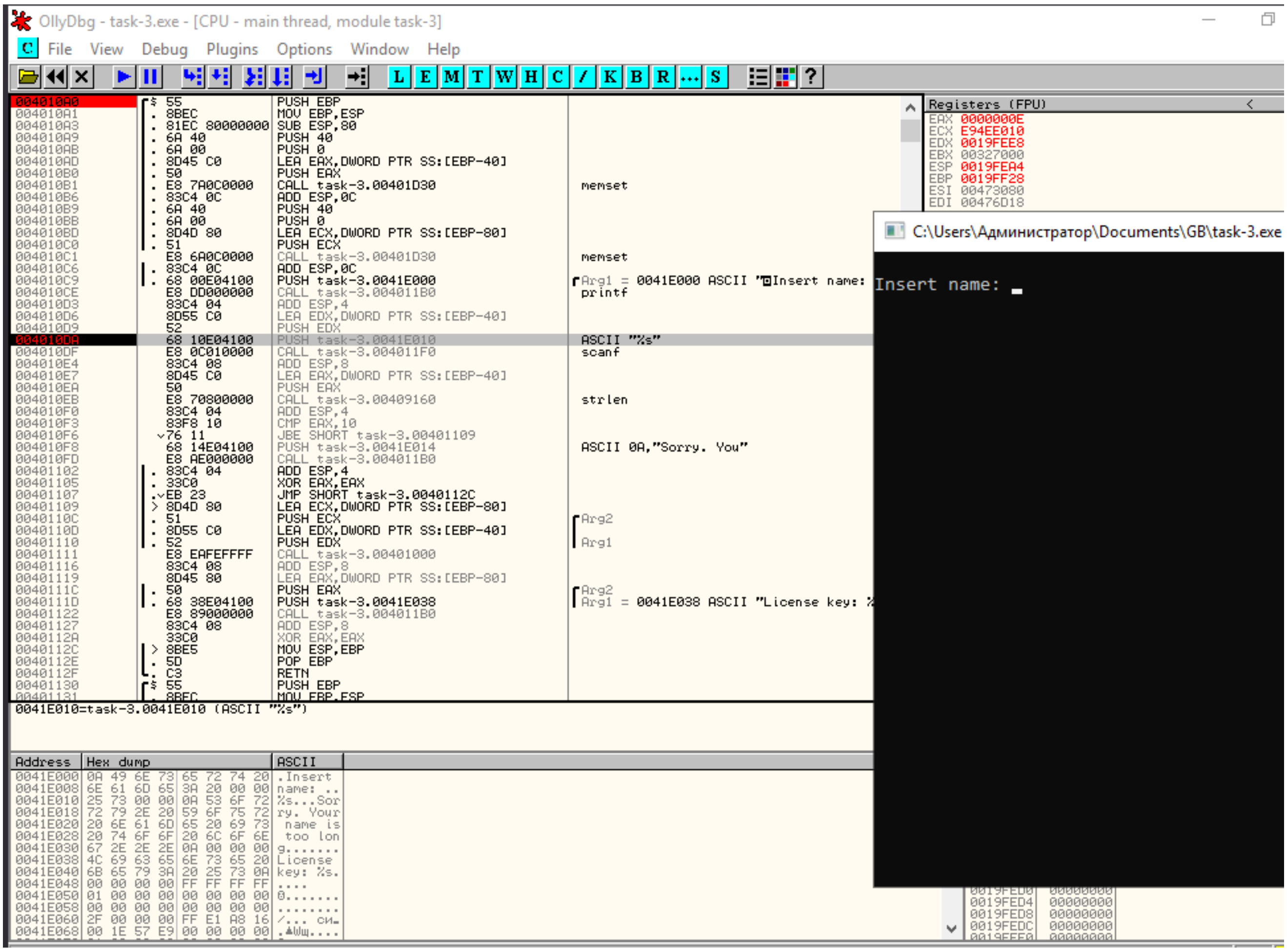
```
Address  Hex dump                 ASCII
0041E000 0A 49 6E 73 65 72 74 20 .Insert
0041E008 6E 61 6D 65 3A 20 00 00 name: ..
0041E010 25 73 00 00 0A 53 6F 72 %s...Sor
0041E018 72 79 2E 20 59 6F 75 72 ry. Your
0041E020 20 6E 61 6D 65 20 69 73  name is
0041E028 20 74 6F 6F 20 6C 6F 6E  too lon
0041E030 67 2E 2E 2E 0A 00 00 00 g.......
```

C:\Users\Администратор\Documents\GB\task-3.exe

```
Insert name:
```



```
004010C1    E8 6A0C0000     CALL task-3.00401D30                 memset
004010C6  . 83C4 0C         ADD ESP,0C
004010C9  . 68 00E04100     PUSH task-3.0041E000                 Arg1 = 0041E000 ASCII "◘Ins
004010CE  . E8 DD000000     CALL task-3.004011B0                 printf
004010D3  . 83C4 04         ADD ESP,4
004010D6    8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
```


OllyDbg - task-3.exe - [CPU - main thread, module task-3]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?

```
004010A0  $ 55              PUSH EBP
004010A1  . 8BEC            MOV EBP,ESP
004010A3  . 81EC 80000000   SUB ESP,80
004010A9  . 6A 40           PUSH 40
004010AB  . 6A 00           PUSH 0
004010AD  . 8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010B0  . 50              PUSH EAX
004010B1  . E8 7A0C0000     CALL task-3.00401D30                memset
004010B6  . 83C4 0C         ADD ESP,0C
004010B9  . 6A 40           PUSH 40
004010BB  . 6A 00           PUSH 0
004010BD  . 8D4D 80         LEA ECX,DWORD PTR SS:[EBP-80]
004010C0  . 51              PUSH ECX
004010C1    E8 6A0C0000     CALL task-3.00401D30                memset
004010C6  . 83C4 0C         ADD ESP,0C
004010C9  . 68 00E04100     PUSH task-3.0041E000                Arg1 = 0041E000 ASCII "◘Insert name:
004010CE    E8 DD000000     CALL task-3.004011B0                printf
004010D3    83C4 04         ADD ESP,4
004010D6    8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
004010D9    52              PUSH EDX
004010DA    68 10E04100     PUSH task-3.0041E010                ASCII "%s"
004010DF    E8 0C010000     CALL task-3.004011F0                scanf
004010E4    83C4 08         ADD ESP,8
004010E7    8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010EA    50              PUSH EAX
004010EB    E8 70800000     CALL task-3.00409160                strlen
004010F0    83C4 04         ADD ESP,4
004010F3    83F8 10         CMP EAX,10
004010F6   v76 11           JBE SHORT task-3.00401109
004010F8    68 14E04100     PUSH task-3.0041E014                ASCII 0A,"Sorry. You"
004010FD    E8 AE000000     CALL task-3.004011B0
00401102  . 83C4 04         ADD ESP,4
00401105  . 33C0            XOR EAX,EAX
00401107  .vEB 23           JMP SHORT task-3.0040112C
00401109  > 8D4D 80         LEA ECX,DWORD PTR SS:[EBP-80]
0040110C  . 51              PUSH ECX                            Arg2
0040110D  . 8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
00401110  . 52              PUSH EDX                            Arg1
00401111    E8 EAFEFFFF     CALL task-3.00401000
00401116    83C4 08         ADD ESP,8
00401119    8D45 80         LEA EAX,DWORD PTR SS:[EBP-80]
0040111C  . 50              PUSH EAX                            Arg2
0040111D  . 68 38E04100     PUSH task-3.0041E038                Arg1 = 0041E038 ASCII "License key: %
00401122    E8 89000000     CALL task-3.004011B0
00401127    83C4 08         ADD ESP,8
0040112A    33C0            XOR EAX,EAX
0040112C  > 8BE5            MOV ESP,EBP
0040112E  . 5D              POP EBP
0040112F  . C3              RETN
00401130  $ 55              PUSH EBP
00401131  . 8BEC            MOV EBP,ESP
0041E010=task-3.0041E010 (ASCII "%s")
```

```
Registers (FPU)
EAX 0000000E
ECX E94EE010
EDX 0019FEE8
EBX 00327000
ESP 0019FEA4
EBP 0019FF28
ESI 00473080
EDI 00476D18
```

C:\Users\Администратор\Documents\GB\task-3.exe

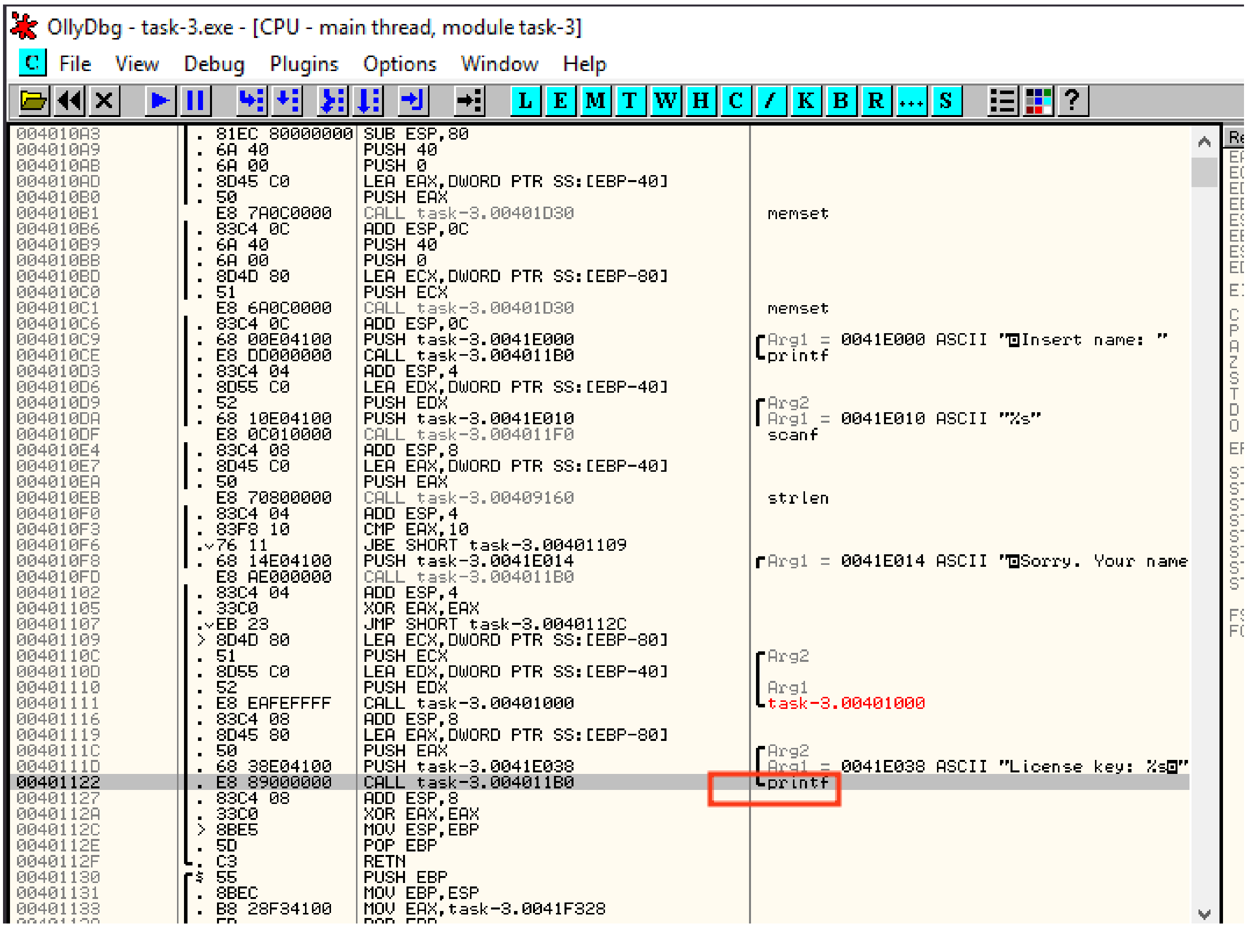
```
Insert name: _
```

```
Address   Hex dump                 ASCII
0041E000  0A 49 6E 73 65 72 74 20  .Insert
0041E008  6E 61 6D 65 3A 20 00 00  name: ..
0041E010  25 73 00 00 0A 53 6F 72  %s...Sor
0041E018  72 79 2E 20 59 6F 75 72  ry. Your
0041E020  20 6E 61 6D 65 20 69 73   name is
0041E028  20 74 6F 6F 20 6C 6F 6E   too lon
0041E030  67 2E 2E 2E 0A 00 00 00  g.......
0041E038  4C 69 63 65 6E 73 65 20  License
0041E040  6B 65 79 3A 20 25 73 0A  key: %s.
0041E048  00 00 00 00 FF FF FF FF  ....
0041E050  01 00 00 00 00 00 00 00  ☺.......
0041E058  00 00 00 00 00 00 00 00  ........
0041E060  2F 00 00 00 FF E1 A8 16  /... си▬
0041E068  00 1E 57 E9 00 00 00 00  .▲Wщ....
```

```
0019FED4  00000000
0019FED8  00000000
0019FEDC  00000000
```

Analyse code

- переименуем функцию в $printf$


OllyDbg - task-3.exe - [CPU - main thread, module task-3]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?

```
004010A3  . 81EC 80000000   SUB ESP,80
004010A9  . 6A 40           PUSH 40
004010AB  . 6A 00           PUSH 0
004010AD  . 8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010B0  . 50              PUSH EAX
004010B1  . E8 7A0C0000     CALL task-3.00401D30                memset
004010B6  . 83C4 0C         ADD ESP,0C
004010B9  . 6A 40           PUSH 40
004010BB  . 6A 00           PUSH 0
004010BD  . 8D4D 80         LEA ECX,DWORD PTR SS:[EBP-80]
004010C0  . 51              PUSH ECX
004010C1  . E8 6A0C0000     CALL task-3.00401D30                memset
004010C6  . 83C4 0C         ADD ESP,0C
004010C9  . 68 00E04100     PUSH task-3.0041E000                Arg1 = 0041E000 ASCII "◘Insert name: "
004010CE  . E8 DD000000     CALL task-3.004011B0                printf
004010D3  . 83C4 04         ADD ESP,4
004010D6  . 8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
004010D9  . 52              PUSH EDX                            Arg2
004010DA  . 68 10E04100     PUSH task-3.0041E010                Arg1 = 0041E010 ASCII "%s"
004010DF    E8 0C010000     CALL task-3.004011F0                scanf
004010E4  . 83C4 08         ADD ESP,8
004010E7  . 8D45 C0         LEA EAX,DWORD PTR SS:[EBP-40]
004010EA  . 50              PUSH EAX
004010EB    E8 70800000     CALL task-3.00409160                strlen
004010F0  . 83C4 04         ADD ESP,4
004010F3  . 83F8 10         CMP EAX,10
004010F6  .v76 11           JBE SHORT task-3.00401109
004010F8  . 68 14E04100     PUSH task-3.0041E014                Arg1 = 0041E014 ASCII "◘Sorry. Your name
004010FD  . E8 AE000000     CALL task-3.004011B0
00401102  . 83C4 04         ADD ESP,4
00401105  . 33C0            XOR EAX,EAX
00401107  .vEB 23           JMP SHORT task-3.0040112C
00401109  > 8D4D 80         LEA ECX,DWORD PTR SS:[EBP-80]
0040110C  . 51              PUSH ECX                            Arg2
0040110D  . 8D55 C0         LEA EDX,DWORD PTR SS:[EBP-40]
00401110  . 52              PUSH EDX                            Arg1
00401111  . E8 EAFEFFFF     CALL task-3.00401000                task-3.00401000
00401116  . 83C4 08         ADD ESP,8
00401119  . 8D45 80         LEA EAX,DWORD PTR SS:[EBP-80]
0040111C  . 50              PUSH EAX                            Arg2
0040111D  . 68 38E04100     PUSH task-3.0041E038                Arg1 = 0041E038 ASCII "License key: %s◘"
00401122  . E8 89000000     CALL task-3.004011B0                printf
00401127  . 83C4 08         ADD ESP,8
0040112A  . 33C0            XOR EAX,EAX
0040112C  > 8BE5            MOV ESP,EBP
0040112E  . 5D              POP EBP
0040112F  . C3              RETN
00401130  $ 55              PUSH EBP
00401131  . 8BEC            MOV EBP,ESP
00401133  . B8 28F34100     MOV EAX,task-3.0041F328
```

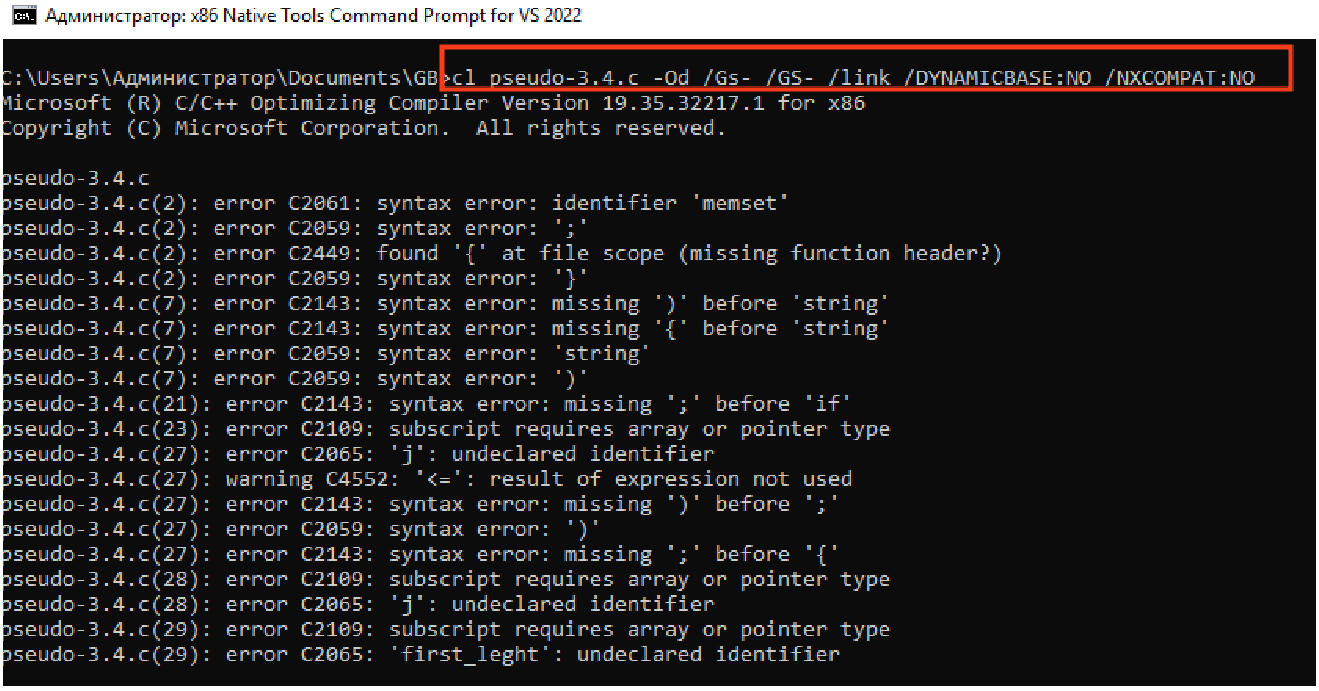
```
//pseudo-3.3.c
// Pseudocode

char buff0[64]
memset{buff0,0,64}

char buff1[64]
memset{buff1,0,64}

printf("Insert name: ")
scanf("%s", buff0)

if (streln(nuff) <=16)
        some_func(buff0, buff1)
        printf("key is %s", buff1)
        return 0
else {
        printf("Sorry")
        return 0;
}

void some_func(buff0, buff1) {

        int i = streln(buff0)
                for(i; i < 8; i++) {
                buff0[i]= 0x61 + i;
        }

        for (int i = 0; j= 0; i <= 16; i++)

                buff1[i] = buff1[j] + 1;
                buff1[i] = 0x41 + i + first_leght;
}
```

Компиляция:

```
cl pseudo-3.3.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
```

Запуск:

```
pseudo-3.3.exe ... (выполняется с ошибками, так как псевдо-код)
```



Администратор: x86 Native Tools Command Prompt for VS 2022

```
C:\Users\Администратор\Documents\GB>cl pseudo-3.4.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
Microsoft (R) C/C++ Optimizing Compiler Version 19.35.32217.1 for x86
Copyright (C) Microsoft Corporation.  All rights reserved.

pseudo-3.4.c
pseudo-3.4.c(2): error C2061: syntax error: identifier 'memset'
pseudo-3.4.c(2): error C2059: syntax error: ';'
pseudo-3.4.c(2): error C2449: found '{' at file scope (missing function header?)
pseudo-3.4.c(2): error C2059: syntax error: '}'
pseudo-3.4.c(7): error C2143: syntax error: missing ')' before 'string'
pseudo-3.4.c(7): error C2143: syntax error: missing '{' before 'string'
pseudo-3.4.c(7): error C2059: syntax error: 'string'
pseudo-3.4.c(7): error C2059: syntax error: ')'
pseudo-3.4.c(21): error C2143: syntax error: missing ';' before 'if'
pseudo-3.4.c(23): error C2109: subscript requires array or pointer type
pseudo-3.4.c(27): error C2065: 'j': undeclared identifier
pseudo-3.4.c(27): warning C4552: '<=': result of expression not used
pseudo-3.4.c(27): error C2143: syntax error: missing ')' before ';'
pseudo-3.4.c(27): error C2059: syntax error: ')'
pseudo-3.4.c(27): error C2143: syntax error: missing ';' before '{'
pseudo-3.4.c(28): error C2109: subscript requires array or pointer type
pseudo-3.4.c(28): error C2065: 'j': undeclared identifier
pseudo-3.4.c(29): error C2109: subscript requires array or pointer type
pseudo-3.4.c(29): error C2065: 'first_leght': undeclared identifier
```

OllyDbg - version 1.10

Quick start - version 1.10

Pop-up menus display only items that apply. **Frequently used menu functions:**

Function	**Window**	**Menu command**	**Shortcut**
Edit memory as binary, ASCII or UNICODE string	Disassembler, Stack Dump	Binary|Edit	Ctrl+E
Undo changes	Disassembler, Dump Registers	Undo selection Undo	Alt+BkSp
Run application	Main	Debug|Run	F9
Run to selection	Disassembler	Breakpoint|Run to selection	F4
Execute till return	Main	Debug|Execute till return	Ctrl+F9
Execute till user code	Main	Debug|Execute till user code	Alt+F9
Set/reset INT3 breakpoint	Disassembler Names, Source	Breakpoint|Toggle Toggle breakpoint	F2
Set/edit conditional INT3 breakpoint	Disassembler Names, Source	Breakpoint|Conditional Conditional breakpoint	Shift+F2
Set/edit conditional logging breakpoint (logs into the Log window)	Disassembler Names, Source	Breakpoint|Conditional log Conditional log breakpoint	Shift+F4

Temporarily disable/restore INT3 breakpoint	Breakpoints	Disable Enable	Space
Set memory breakpoint (only one is allowed)	Disassembler, Dump	Breakpoint|Memory, on access Breakpoint|Memory, on write	
Remove memory breakpoint	Disassembler, Dump	Breakpoint|Remove memory breakpoint	
Set hardware breakpoint (ME/NT/2000 only)	Disassembler, Dump	Breakpoint|Hardware (select type and size!)	
Remove hardware breakpoint	Main	Debug|Hardware breakpoints	
Set single-short break on access to memory block (NT/2000 only)	Memory	Set break-on-access	F2
Set break on module, thread, debug string	Options	Events	
Set new origin	Disassembler	New origin here	
Display list of all symbolic names	Disassembler, Dump Modules	Search for|Name (label) View names	Ctrl+N
Context-sensitive help (requires external help file!)	Disassembler, Names	Help on symbolic name	Ctrl+F1
Find all references in code to selected address range	Disassembler Dump	Find references to|Command Find references	Ctrl+R
Find all references in code to the constant	Disassembler	Find references to|Constant Search for|All constants	
Search whole allocated memory	Memory	Search Search next	Ctrl+L
Go to address or value of expression	Disassembler Dump	Go to|Expression Go to expression	Ctrl+G
Go to previous address/run trace item	Disassembler	Go to|Previous	Minus
Go to next address/run trace item	Disassembler	Go to|Next	Plus
Go to previous procedure	Disassembler	Go to|Previous procedure	Ctrl+Minus
Go to next procedure	Disassembler	Go to|Next procedure	Ctrl+Plus

View executable file	Disassembler, Dump, Modules	View|Executable file	
Copy changes to executable file	Disassembler	Copy to executable file	
Analyse executable code	Disassembler	Analysis|Analyse code	Ctrl+A
Scan object files and libraries	Disassembler	Scan object files	Ctrl+O
View resources	Modules, Memory	View all resources View resource strings	
Suspend/resume thread	Threads	Suspend Resume	
Display relative addresses	Disassembler, Dump, Stack	Doubleclick address	
Copy	Most of windows	Copy to clipboard	Ctrl+C

Frequently used global shortcuts:

Ctrl+F2	Restart program
Alt+F2	Close program
F3	Open new program
F5	Maximize/restore active window
Alt+F5	Make OllyDbg topmost
F7	Step into (entering functions)
Ctrl+F7	Animate into (entering functions)
F8	Step over (executing function calls at once)
Ctrl+F8	Animate over (executing function calls at once)
F9	Run
Shift+F9	Pass exception to standard handler and run
Ctrl+F9	Execute till return
Alt+F9	Execute till user code
Ctrl+F11	Trace into
F12	Pause
Ctrl+F12	Trace over
Alt+B	Open Breakpoints window
Alt+C	Open CPU window
Alt+E	Open Modules window
Alt+L	Open Log window
Alt+M	Open Memory window
Alt+O	Open Options dialog

|**Ctrl+T**|Set condition to pause Run trace|
|**Alt+X**|Close OllyDbg|

Frequently used Disasembler shortcuts:

F2	Toggle breakpoint
Shift+F2	Set conditional breakpoint
F4	Run to selection
Alt+F7	Go to previous reference
Alt+F8	Go to next reference
Ctrl+A	Analyse code
Ctrl+B	Start binary search
Ctrl+C	Copy selection to clipboard
Ctrl+E	Edit selection in binary format
Ctrl+F	Search for a command
Ctrl+G	Follow expression
Ctrl+J	Show list of jumps to selected line
Ctrl+K	View call tree
Ctrl+L	Repeat last search
Ctrl+N	Open list of labels (names)
Ctrl+O	Scan object files
Ctrl+R	Find references to selected command
Ctrl+S	Search for a sequence of commands
Asterisk (*)	Origin
Enter	Follow jump or call
Plus (+)	Go to next location/next run trace item
Minus (-)	Go to previous location/previous run trace item
Space ( )	Assemble
Colon (:)	Add label
Semicolon (;)	Add comment

*Выполнил: AndreiM