rdi = rbx = input 0x4f8fe8

```
0000000000402228 <phase_4>:
  402228:	f3 0f 1e fa          	endbr64
  40222c:	55                   	push   rbp
  40222d:	53                   	push   rbx
  40222e:	48 83 ec 18          	sub    rsp,0x18
  402232:	48 89 fb             	mov    rbx,rdi   rbx = 1er argumento
  402235:	64 48 8b 04 25 28 00 	mov    rax,QWORD PTR fs:0x28
  40223c:	00 00
  40223e:	48 89 44 24 08       	mov    QWORD PTR [rsp+0x8],rax
  402243:	31 c0                	xor    eax,eax   rax = 0
  402245:	48 8d 6c 24 01       	lea    rbp,[rsp+0x1]   rbp = buffer en la pila
  40224a:	48 89 ea             	mov    rdx,rbp
  40224d:	48 8d 35 02 c1 0d 00 	lea    rsi,[rip+0xdc102]        # 4de356 <blanks+0x56>
  402254:	e8 17 08 01 00       	call   412a70 <__isoc99_sscanf>
  402259:	89 c2                	mov    edx,eax   rdx = resultado de scanf(rbx)
  40225b:	48 c7 c1 ff ff ff ff 	mov    rcx,0xffffffffffffffff   rcx = -1
  402262:	b8 00 00 00 00       	mov    eax,0x0   a = 0
  402267:	48 89 ef             	mov    rdi,rbp   rdi = sp+1   ACA ESTÁ EL INPUT
                                                                 ↖input
  40226a:	f2 ae                	repnz scas al,BYTE PTR es:[rdi]   busca un byte en una cadena
  40226c:	48 f7 d1             	not    rcx          loop x 8
  40226f:	48 83 e9 01          	sub    rcx,0x1
  402273:	83 fa 01             	cmp    edx,0x1   → es 1 elemento de entrada
  402276:	75 06                	jne    40227e <phase_4+0x56>
  402278:	48 83 f9 06          	cmp    rcx,0x6   → el tamaño del buffer leído
  40227c:	74 05                	je     402283 <phase_4+0x5b>
  40227e:	e8 cf 02 00 00       	call   402552 <explode_bomb>   tamaño = 6   o sino BOMBA
  402283:	b8 00 00 00 00       	mov    eax,0x0
  402288:	48 8d 0d 51 2f 0c 00 	lea    rcx,[rip+0xc2f51]        # 4c51e0 <array.3482>
  40228f:	0f b6 14 03          	movzx  edx,BYTE PTR [rbx+rax*1]
  402293:	83 e2 0f             	and    edx,0xf
  402296:	0f b6 14 11          	movzx  edx,BYTE PTR [rcx+rdx*1]   loop x6
  40229a:	88 54 04 01          	mov    BYTE PTR [rsp+rax*1+0x1],dl
  40229e:	48 83 c0 01          	add    rax,0x1
  4022a2:	48 83 f8 06          	cmp    rax,0x6
  4022a6:	75 e7                	jne    40228f <phase_4+0x67>
  4022a8:	c6 44 24 07 00       	mov    BYTE PTR [rsp+0x7],0x0   → 0 al 8vo lugar de la pila
  4022ad:	48 8d 74 24 01       	lea    rsi,[rsp+0x1]   rsi apunta a desp de rbp   cmjbck
  4022b2:	b9 07 00 00 00       	mov    ecx,0x7   c = 7
  4022b7:	48 8d 3d 0d 2f 0c 00 	lea    rdi,[rip+0xc2f0d]   valor esperado lechon   # 4c51cb <_IO_stdin_used+0x1cb>
  4022be:	f3 a6                	repz cmps BYTE PTR ds:[rsi],BYTE PTR es:[rdi]   compara   ip
  4022c0:	0f 97 c0             	seta   al   → 1 si rsi > rdi , 0 si =
  4022c3:	1c 00                	sbb    al,0x0
  4022c5:	84 c0                	test   al,al
  4022c7:	75 17                	jne    4022e0 <phase_4+0xb8>
  4022c9:	48 8b 44 24 08       	mov    rax,QWORD PTR [rsp+0x8]
  4022ce:	64 48 33 04 25 28 00 	xor    rax,QWORD PTR fs:0x28
  4022d5:	00 00
  4022d7:	75 0e                	jne    4022e7 <phase_4+0xbf>
  4022d9:	48 83 c4 18          	add    rsp,0x18
  4022dd:	5b                   	pop    rbx
  4022de:	5d                   	pop    rbp
  4022df:	c3                   	ret
  4022e0:	e8 6d 02 00 00       	call   402552 <explode_bomb>
  4022e5:	eb e2                	jmp    4022c9 <phase_4+0xa1>
  4022e7:	e8 54 55 05 00       	call   457840 <__stack_chk_fail>
```

$\boxed{rsi = rdi}$

byte por byte
pero me elimina el primer ch. de input

49	50	53	57	54	51
1	2	5	9	6	3

rdi = 1 2 5 9 6 3

lechon -1
echon -2
chon -3
hon -4
on -5
n -6
-7
-8

rdi

pero en rbp
está la palabra
completa

-8 frena cuando rcx = -8

8 → 00001000
¬8 + 1 = 1111 0 111 + 1
= 11111000

desp = not(-8) = 0000111 = 7

desp resta 1 = 6

rax	1	rcx
	e	101
	g	103
	m	109
	c	99
	f	102
	a	97
	i	105
	j	106
	o	111
	p	112
	n	110
	h	104

rsi gggggg

rdi lechon

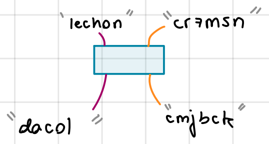
rsi : d a c o l k

v d a c o l k sp

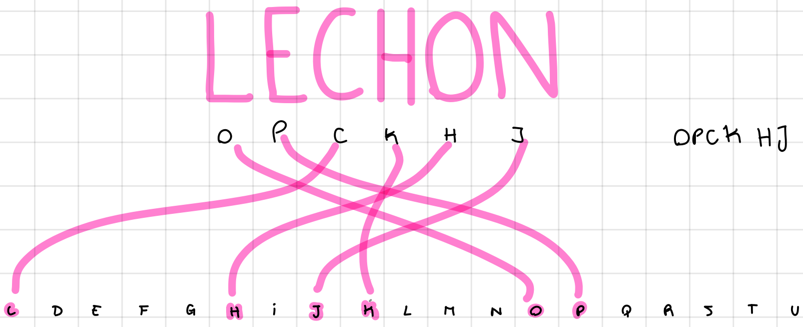
ADi	L	E	C	H	O	N
rdx	108 1101100	101 01100101	99 01100011	104 01101000	111 01101111	110 01101110
and (rdx, OxF)	↓ 12	↓ 5	↓ 3	↓ 8	↓ 15	↓ 14
rcx + rdx	↓ 100	↓ 97	↓ 99	↓ 111	↓ 108	↓ 107
	D	A	C	O	L	K

o

Asi necesito que se transforme en lechōn



A	B	C	D	e	F	G	H	i	J	K	L	M	N	O	P	Q	R	S	T	U
97		99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	14	15	16	17
A		C	D																	



A	B	C	D	E	F	G	H	i	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	x	y	z
G	M	C	F	A	i	J	O	P	N	H	D	B	K	L	E	G	M								
103	109	99	102	97	105	106		112	110			98			101	103	109								