Bomb cmu15:213

Phase1

```
Border relations with Canada have never been better.
```

Phase2

```
1 2 4 8 16 32
```

phase_3

```
6 682
```

Phase4

```
7 0
```

Phase5

```
ionefg
```

```
0x0000000000401062 <+0>: push   %rbx
0x0000000000401063 <+1>: sub    $0x20,%rsp
0x0000000000401067 <+5>: mov    %rdi,%rbx
0x000000000040106a <+8>: mov    %fs:0x28,%rax
0x0000000000401073 <+17>:    mov    %rax,0x18(%rsp)
0x0000000000401078 <+22>:    xor    %eax,%eax
0x000000000040107a <+24>:    call   0x40131b <string_length>
0x000000000040107f <+29>:    cmp    $0x6,%eax
0x0000000000401082 <+32>:    je     0x4010d2 <phase_5+112>
0x0000000000401084 <+34>:    call   0x40143a <explode_bomb>
0x0000000000401089 <+39>:    jmp    0x4010d2 <phase_5+112>


-----------------------------------------------------------------
#cycle
0x000000000040108b <+41>:    movzbl (%rbx,%rax,1),%ecx #move 1st char to
ecx
0x000000000040108f <+45>:    mov    %cl,(%rsp)
0x0000000000401092 <+48>:    mov    (%rsp),%rdx #move nth char to rdx
0x0000000000401096 <+52>:    and    $0xf,%edx  #1111& value in edx(ASCI of
char)
0x0000000000401099 <+55>:    movzbl 0x4024b0(%rdx),%edx
!!!!! *0x4024b0
maduiersnfotvbylSo   f l  y  e r s   a-97-110 0001 z-122-111 1010
                     9 15 14 5 6 7
1101001 1101111 1101110 1100101 1100110 1100111
105 i    111 o    110 n    101 e    102 f     103 g
honefg
0x00000000004010a0 <+62>:    mov    %dl,0x10(%rsp,%rax,1) #move %rdx to
(%rsp+%rax*1+0x10)
0x00000000004010a4 <+66>:    add    $0x1,%rax
0x00000000004010a8 <+70>:    cmp    $0x6,%rax
0x00000000004010ac <+74>:    jne    0x40108b <phase_5+41>

char[] s =maduiersnfotvbylSo
```

```
for(i=0,i<6,i++){
    phase_5+41
    char ch = input[6]
    s[ch]->%rsp+i*1+0x10
    phase_5+74
}
```

```
0x00000000004010ae <+76>:    movb   $0x0,0x16(%rsp)
0x00000000004010b3 <+81>:    mov    $0x40245e,%esi
0x40245e flyers
0x00000000004010b8 <+86>:    lea    0x10(%rsp),%rdi
0x00000000004010bd <+91>:    call   0x401338 <strings_not_equal>
0x00000000004010c2 <+96>:    test   %eax,%eax
0x00000000004010c4 <+98>:    je     0x4010d9 <phase_5+119>
0x00000000004010c6 <+100>:   call   0x40143a <explode_bomb>
0x00000000004010cb <+105>:   nopl   0x0(%rax,%rax,1)
0x00000000004010d0 <+110>:   jmp    0x4010d9 <phase_5+119

0x00000000004010d2 <+112>:   mov    $0x0,%eax
0x00000000004010d7 <+117>:   jmp    0x40108b <phase_5+41>
0x00000000004010d9 <+119>:   mov    0x18(%rsp),%rax
0x00000000004010de <+124>:   xor    %fs:0x28,%rax
0x00000000004010e7 <+133>:   je     0x4010ee <phase_5+140>
0x00000000004010e9 <+135>:   call   0x400b30 <__stack_chk_fail@plt>
0x00000000004010ee <+140>:   add    $0x20,%rsp
0x00000000004010f2 <+144>:   pop    %rbx
0x00000000004010f3 <+145>:   ret
```

Phase6

4 3 2 1 6 5

```
   0x00000000004010f4 <+0>:    push   %r14
   0x00000000004010f6 <+2>:    push   %r13
   0x00000000004010f8 <+4>:    push   %r12
   0x00000000004010fa <+6>:    push   %rbp
   0x00000000004010fb <+7>:    push   %rbx
   0x00000000004010fc <+8>:    sub    $0x50,%rsp
   0x0000000000401100 <+12>:    mov    %rsp,%r13    #r13=rsp
   0x0000000000401103 <+15>:    mov    %rsp,%rsi    #rsi=rsp
   0x0000000000401106 <+18>:    call   0x40145c <read_six_numbers>
   0x000000000040110b <+23>:    mov    %rsp,%r14     #r14=rsp
 **0x000000000040110e <+26>:    mov    $0x0,%r12d *    #r12d=0
```

```
   0x0000000000401114 <+32>:    mov    %r13,%rbp     rbp=r13= address of argum 1
   0x0000000000401117 <+35>:    mov    0x0(%r13),%eax   eax=argum 1
   0x000000000040111b <+39>:    sub    $0x1,%eax   eax-1   argum1-1
   0x000000000040111e <+42>:    cmp    $0x5,%eax
   0x0000000000401121 <+45>:    jbe    0x401128 <phase_6+52> so argum1-1<=5 argums
<=6
   0x0000000000401123 <+47>:    call   0x40143a <explode_bomb>
```

```
 **0x0000000000401128 <+52>:    add    $0x1,%r12d #r12d=0+1=1**
   0x000000000040112c <+56>:    cmp    $0x6,%r12d  #loop 6 times *
   0x0000000000401130 <+60>:    je     0x401153 <phase_6+95>
   0x0000000000401132 <+62>:    mov    %r12d,%ebx  ebx=1
   ____________________________________________________________
   0x0000000000401135 <+65>:    movslq %ebx,%rax    rax=1
   0x0000000000401138 <+68>:    mov    (%rsp,%rax,4),%eax  #move next argum to eax
   0x000000000040113b <+71>:    cmp    %eax,0x0(%rbp)## rbp 1st argum  argums not
equal to current one
   0x000000000040113e <+74>:    jne    0x401145 <phase_6+81>
   0x0000000000401140 <+76>:    call   0x40143a <explode_bomb>
   0x0000000000401145 <+81>:    add    $0x1,%ebx  ebx+1
   0x0000000000401148 <+84>:    cmp    $0x5,%ebx
   ____________________________________________________________
   0x000000000040114b <+87>:    jle    0x401135 <phase_6+65> ebx <=5
   0x000000000040114d <+89>:    add    $0x4,%r13
   0x0000000000401151 <+93>:    jmp    0x401114 <phase_6+32>

for (i=0,i<6,i++){
    if num[i]> 6 explode
    for(j=i,j<6,j++){
    num[i]!=num[j]
}

}

six num, and every num less or qual than 6 ,and all num not qual!!!!!

   ____________________________________________

   0x0000000000401153 <+95>:    lea    0x18(%rsp),%rsi
   0x0000000000401158 <+100>:    mov    %r14,%rax  #rax=r14 address of num1
   0x000000000040115b <+103>:    mov    $0x7,%ecx  #ecx=7

   0x0000000000401160 <+108>:    mov    %ecx,%edx  #edx=ecx=7
   0x0000000000401162 <+110>:    sub    (%rax),%edx  #edx=7- current num
   0x0000000000401164 <+112>:    mov    %edx,(%rax)  #num=7-num
   0x0000000000401166 <+114>:    add    $0x4,%rax  #rax address of next num
   0x000000000040116a <+118>:    cmp    %rsi,%rax
   0x000000000040116d <+121>:    jne    0x401160 <phase_6+108>
   ##loop six times num =7-num


   0x000000000040116f <+123>:    mov    $0x0,%esi     #esi=0
   0x0000000000401174 <+128>:    jmp    0x401197 <phase_6+163>

   0x0000000000401176 <+130>:    mov    0x8(%rdx),%rdx
   0x000000000040117a <+134>:    add    $0x1,%eax    eax=1+1=2
   0x000000000040117d <+137>:    cmp    %ecx,%eax      ecx=7-num1 eax=2
   ##loop when eax=7-num1  so rdx=address of node[eax]

   0x000000000040117f <+139>:    jne    0x401176 <phase_6+130

   0x0000000000401181 <+141>:    jmp    0x401188 <phase_6+148>
   0x0000000000401183 <+143>:    mov    $0x6032d0,%edx   #edx=332
   0x0000000000401188 <+148>:    mov    %rdx,0x20(%rsp,%rsi,2)
   0x000000000040118d <+153>:    add    $0x4,%rsi   rsi=4
   0x0000000000401191 <+157>:    cmp    $0x18,%rsi   rsi  0x18
   0x0000000000401195 <+161>:    je     0x4011ab <phase_6+183>

   0x0000000000401197 <+163>:    mov    (%rsp,%rsi,1), %ecx  #ecx=7-num1
   0x000000000040119a <+166>:    cmp    $0x1,%ecx     #compare 7-num1  with 1
   0x000000000040119d <+169>:    jle    0x401183 <phase_6+143>  # num1<=1
   0x000000000040119f <+171>:    mov    $0x1,%eax         #eax=1
   0x00000000004011a4 <+176>:    mov    $0x6032d0,%edx    #edx=332
   0x00000000004011a9 <+181>:    jmp    0x401176 <phase_6+130>
```

```
phase6+130

                                                          . . . . . .

                                                        node[7-input2]
address of node[7-nthinput]->somewhere in stack (rsp+0x20)  node[7-input1]



phase6+181

   0x00000000004011ab <+183>:    mov    0x20(%rsp),%rbx  ##rbx= address of 1st
node6
   0x00000000004011b0 <+188>:    lea    0x28(%rsp),%rax  ##rax= address of  2nd
node5
   0x00000000004011b5 <+193>:    lea    0x50(%rsp),%rsi
   0x00000000004011ba <+198>:    mov    %rbx,%rcx   ##rcx= addrss of 1st node6

   0x00000000004011bd <+201>:    mov    (%rax),%rdx  ##rdx=address of 2nd node5
   0x00000000004011c0 <+204>:    mov    %rdx,0x8(%rcx)  ##move address of 2nd node
to node*next of 1stnode
   0x00000000004011c4 <+208>:    add    $0x8,%rax     ##rax=address of 3rd node
   0x00000000004011c8 <+212>:    cmp    %rsi,%rax    ##rsi is end of linked list
   0x00000000004011cb <+215>:    je     0x4011d2 <phase_6+222>
   0x00000000004011cd <+217>:    mov    %rdx,%rcx   ##rcx=address of 2nd node
   0x00000000004011d0 <+220>:    jmp    0x4011bd <phase_6+201>

   ##re-connect of new linked list  order by 7-imputNum
   0x00000000004011d2 <+222>:    movq   $0x0,0x8(%rdx)  set the node*next of last
node to null

   0x00000000004011da <+230>:    mov    $0x5,%ebp  #ebp=5
   0x00000000004011df <+235>:    mov    0x8(%rbx),%rax    rax = address of 2nd node
   0x00000000004011e3 <+239>:    mov    (%rax),%eax       eax=nodeVal of 2nd node
   0x00000000004011e5 <+241>:    cmp    %eax,(%rbx)       (rbx) =nodeVal of 1st
node
   0x00000000004011e7 <+243>:    jge    0x4011ee <phase_6+250> 1st node >= 2nd node
   0x00000000004011e9 <+245>:    call   0x40143a <explode_bomb>
   0x00000000004011ee <+250>:    mov    0x8(%rbx),%rbx     rbx=address of 2nd node
   0x00000000004011f2 <+254>:    sub    $0x1,%ebp       #ebp-1=4
   0x00000000004011f5 <+257>:    jne    0x4011df <phase_6+235>
  ## loop to compare ajacent nodeVal so new list is decreasing
   0x00000000004011f7 <+259>:    add    $0x50,%rsp
   0x00000000004011fb <+263>:    pop    %rbx
   0x00000000004011fc <+264>:    pop    %rbp
   0x00000000004011fd <+265>:    pop    %r12
   0x00000000004011ff <+267>:    pop    %r13
   0x0000000000401201 <+269>:    pop    %r14
   0x0000000000401203 <+271>:    ret



 linked list of 6 nodes

 struct node{
  int nodeVal; #like 332
  int nodeNum;#which node node1 node2...
  struct node* next
}

node1  node2 node3  node4  node5  node6
332    168    924    691   477     443
924    691    477    443   332     168
node3  node4  node5  node6 node1  node2
7-3     7-4   7-5    7-6   7-1    7-2
4 3 2 1 6 5
```