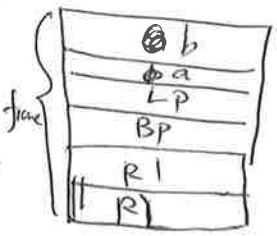
Problem 1. Consider the following implementation of an algorithm for finding the greatest common divisor of two integers:

```
int gcd(int a,int b) {
  if (a == b) return a;
  if (a > b) return gcd(a-b,b);
  return gcd(a,b-a);
}
```

The C compiler has compiled this procedure into the following code for an ~~unpipelined~~ Beta processor:




```
gcd:
        PUSH (LP)
        PUSH (BP)
        MOVE (SP, BP)
        PUSH (R1)
        PUSH (R2)
        LD (BP, -12, R0)
        LD (BP, -16, R1)
        CMPEQ (R0, R1, R2)
        BT (R2, L1)
        CMPLE (R0, R1, R2)
        BT (R2, L2)
        PUSH (R1)
        SUB (R0, R1, R2)
        PUSH (R2)
        BR (gcd, LP)
        DEALLOCATE (2)
        BR (L1)
L2:
        SUB (R1, R0, R2)
        PUSH (R2)
        PUSH (R0)
        BR (gcd, LP)
        DEALLOCATE (2)
L1:
        POP (R2)
        POP (R1)
        MOVE (BP, SP)
        POP (BP)
        POP (LP)
        JMP (LP)
```

load a
load b
a==b ?
return a
a<b ?
~~return~~ gcd
b-a
b'
a'
call
remove arguments
a-b
b'
a'
call
remove args
restore
return.

load argument b.

A. The program above contains the instruction LD(BP,-16,R1). Explain what the compiler was trying to do when it generated this instruction.

B. What are the contents of the memory location holding the instruction BR(L1)?

PUSH ⇒ 2
ID, 31, 31, -7 → 01 11 01 11111 11111 0000000000000111
(PC+4+4X=PC+4+8
↳ X+1=8

C. When the instruction labeled "L1" is executed, what is the best characterization of the contents of R0?

return

D. Looking at the code, a student suggests that both DEALLOCATE instructions could be eliminated since deallocation is performed

implicitly by the MOVE(BP,SP) instruction in the exit sequence. After calling gcd, would it be possible to tell if the DEALLOCATE instructions had been removed? Yes. R2 R1 change

E. How many words of stack are needed to execute gcd(24,16)? Don't forget to include the stack space occupied by the arguments in the initial call. 24,16 → 8,16 → 8,8 3 frames. → 18

F. During execution of gcd(28,70), the Beta processor is halted and the contents of portions of the stack are found to contain the following:

```
        ????
        0x00000594
        0x00001234
        0x00000046
        0x0000002A
        0x0000000E
        0x0000001C
        0x00000594
        0x0000124C
BP-->   0x0000002A
        0x0000000E
SP-->   0x00001254
        0x0000000E
```

What is the value of the second argument ("b") to the current call to gcd? 28

G. What is the value in the BP register at the time the stack snapshot was taken? old BP + fram = 124C + 6x4.

H. What is the correct value for "???" above? → 28 = 0x0000001C

28,70 → 28,42 → 28,14 →

I. What is the address of the POP(R2) instruction? gcd: LP 0x594 ; POP(R2): L1

J. At the time the stack snapshot was taken, what is the significance of the value 0x1254 in the location at <SP>? nothing, not written

just add words between L1 and gcd.

K. The stack snapshot was taken just after the execution of a particular instruction. Could the snapshot have been taken just after the execution of the PUSH(R1) instruction near the beginning of the gcd procedure?

no it is just taken after PUSH(R2).