## Q1 Min Function

min (int, int, int)

```
min:
        push    ebp                 ; Save the old ebp
        mov     ebp,esp             ; Move the stack pointer to the ebp
        sub     esp, 4              ; Clear space for the local variable v

        mov     eax, [ebp + 8]      ; eax = a
        mov     [ebp - 4], eax      ; v = eax

        ;Start of the first if statement
        mov     eax, [ebp +12]      ; eax = b
        cmp     eax, [ebp - 4]      ; Compare b and v
        jge     min0                ; If b is greater, branch
        mov     eax , [ebp +12]     ; eax = b
        mov     [ebp - 4], eax      ; v = eax

min0:
        ;Start of second if statement
        mov     eax, [ebp +16]      ; eax = c
        cmp     eax, [ebp-4]        ; compare c and v
        jge     min1                ; If c is greater, branch
        mov     eax, [ebp +16]      ; eax = c
        mov     [ebp-4], eax        ; v = eax

min1:
        ; Returning v
        mov   eax, [ebp-4]          ; eax = v , eax is returned from the function

        ; Closing the enviroment
        mov   esp, ebp              ; Point the stack point back to the ebp
        pop   ebp                   ; pop the old ebp off the stack
        ret   0
```

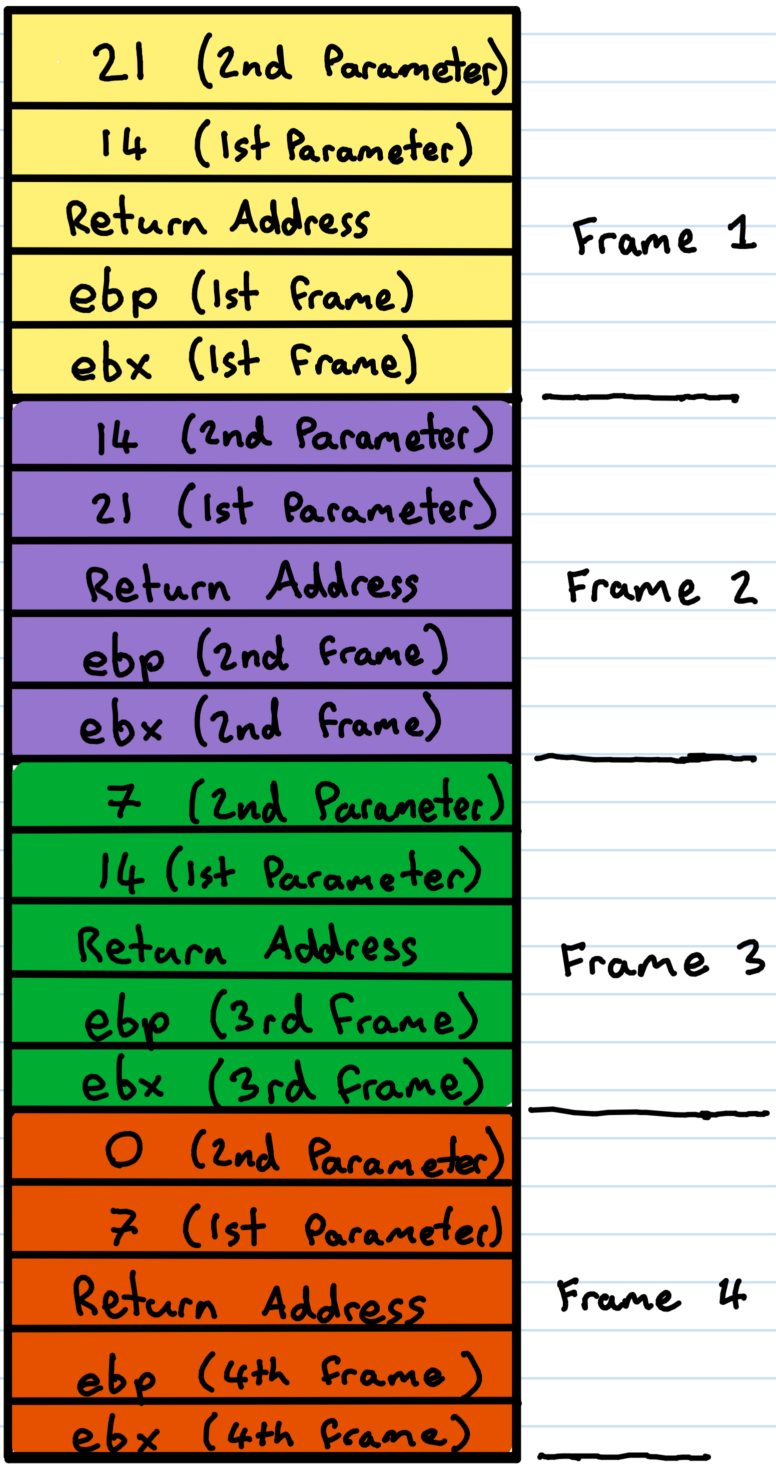
## Q1 p Function p(int, int, int, int)

```
p:
    ;Storing the enviroment
    push   ebp                 ; Save original frame pointer
    mov    ebp,esp             ; Move stack pointer into frame pointer
    sub    esp, 4              ; Make space for Local variable v

    ;Push the parameters for min()
    push   [ebp+12]            ; push j to the stack
    push   [ebp+8]             ; push i to the stack
    push   g                   ; push g to the stack
    call   min                 ; call the function min
    mov    [ebp-4],eax         ; move the result into Local variable
    add    esp, 12             ; pop the parameters off the stack

    ;Push the parameter for min()
    push   [ebp+20]            ; push l to the stack
    push   [ebp+16]            ; push k to the stack
    push   [ebp-4]             ; push v to the stack
    call   min                 ; call the min function
    add    esp, 12             ; pop the parameters off

    ;Closing the Function
    mov    esp, ebp
    pop    ebp
    ret    0
```

# Q1 gcd Function    gcd(int, int)

```
gcd:
        ;Store the enviroment
        push    ebp                 ; Save old ebp
        mov     ebp, esp            ; Move stack pointer to ebp
        push    ebx                 ; Save ebx

        ;First if statement
        mov     eax, [ebp + 12]     ; eax = b
        test    eax, eax            ; test if eax is 0
        jne     notZero             ; Branch if not zero

        ;Return a if b is 0
        mov     eax, [ebp + 8]      ; eax = a
        pop     ebx                 ; Return ebx
        mov     esp, ebp            ; Restore stack pointer
        pop     ebp                 ; Return old ebp
        ret     0

notZero:
        ;Segment calling recursive function
        mov     eax, [ebp + 8]      ; eax = a
        and     edx, 0              ; clear edx
        mov     ebx, [ebp+12]       ; ebx = b
        div     ebx                 ;(eax = ebx/eax) (edx = ebx % eax)
        push    edx                 ; push a%b
        push    ebx                 ; push a
        call    gcd                 ; call function with new params
        add     esp, 8              ; pop the two params

        ;Return result and close function
        pop     ebx
        mov     esp, ebp            ; Move the ebp to the stack pointer
        pop     ebp                 ; Return the old frame pointer
```

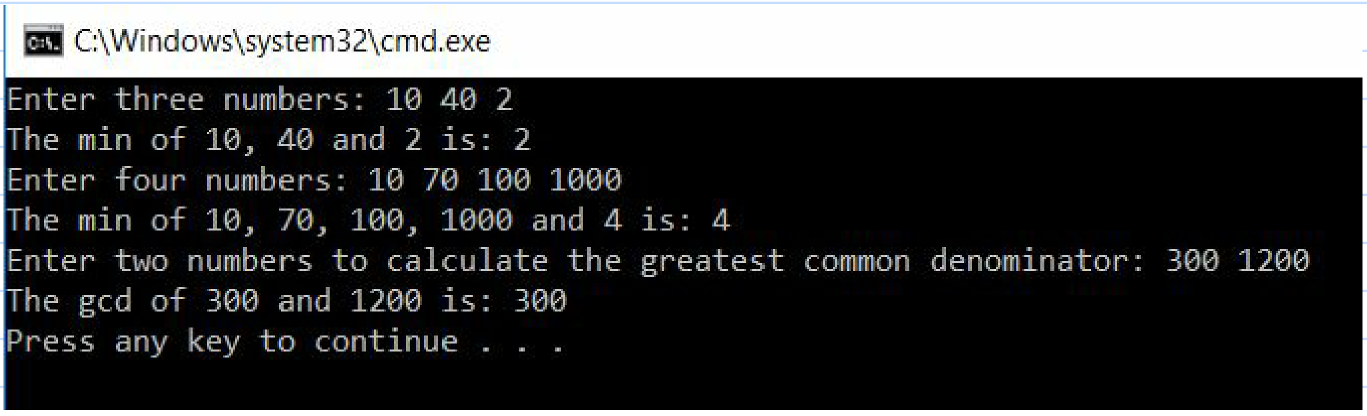
## Q2 Stack Frame



| | |
|---|---|
| gcd (14, 21) | Frame 1 |
| gcd (21, 14) | Frame 2 |
| gcd (14, 7) | Frame 3 |
| gcd (7, 0) | Frame 4 |

As b = 0,
the function returns

## Q3 Console Printout


C:\Windows\system32\cmd.exe

```
Enter three numbers: 10 40 2
The min of 10, 40 and 2 is: 2
Enter four numbers: 10 70 100 1000
The min of 10, 70, 100, 1000 and 4 is: 4
Enter two numbers to calculate the greatest common denominator: 300 1200
The gcd of 300 and 1200 is: 300
Press any key to continue . . .
```