

VU LEARNING HUB

CS401P Assignment 1 Solution

Fall 2026

📢 Join our official WhatsApp Channel for instant VU updates, past papers & assignment solutions!

➡ **https://whatsapp.com/channel/0029VbDOzXMC6ZvZBUbW8k2n**

💬 Questions? Chat with us on WhatsApp for quick help & guidance!

➡ **https://wa.me/923135320064**

🌐 Visit our website for more past papers, notes & assignment solutions!

➡ **https://vulearninghub.xyz**

CS401P — Computer Architecture and Assembly Language Programming (Practical) — Assignment No. 1, Fall 2026

```
; CS401P Assignment 1 - Fall 2026
; Task 1: Split the VUID into prefix, LHS and RHS
; Task 2: GCD of LHS and RHS in BX (0xFFFF if either is zero)
; Replace BC123456789 with your own VUID before submitting.

org 100h
    jmp start

section .data
vuid    db 'BC123456789', 0   ; example VUID: 2 letters + 9 digits
prefix  db '  ', 0            ; 2-letter prefix
LHS     db '    ', 0          ; first 4 digits (as text)
RHS     db '     ', 0         ; remaining 5 digits (as text)
LHS_VAL dw 0                  ; first 4 digits (as number)
RHS_VAL dw 0                  ; remaining 5 digits (as number)

section .text
start:
    ; ---- Task 1: split the VUID ----
    mov si, vuid
    mov di, prefix
    mov cx, 2
copy_prefix:
    mov al, [si]
    mov [di], al
    inc si
    inc di
    loop copy_prefix

    mov di, LHS
    mov cx, 4
copy_lhs:
    mov al, [si]
    mov [di], al
    inc si
    inc di
    loop copy_lhs

    mov di, RHS
    mov cx, 5
copy_rhs:
    mov al, [si]
    mov [di], al
    inc si
```

```
    inc di
    loop copy_rhs

    ; ---- Convert LHS text to a number ----
    mov si, LHS
    call str_to_num
    mov [LHS_VAL], ax

    ; ---- Convert RHS text to a number ----
    mov si, RHS
    call str_to_num
    mov [RHS_VAL], ax

    ; ---- Task 2: GCD by repeated division ----
    mov ax, [LHS_VAL]
    mov bx, [RHS_VAL]
    cmp ax, 0
    je zero_case
    cmp bx, 0
    je zero_case
gcd_loop:
    cmp ax, bx
    jae ax_ge_bx
    xchg ax, bx
ax_ge_bx:
    xor dx, dx
    div bx              ; AX = quotient, DX = remainder
    mov ax, bx
    mov bx, dx
    cmp bx, 0
    jne gcd_loop
    mov bx, ax          ; GCD in BX
    jmp done
zero_case:
    mov bx, 0FFFFh
done:
    mov ax, 4C00h
    int 21h

; Convert null-terminated digit string at SI to a number in AX
str_to_num:
    xor ax, ax
    xor cx, cx
.next_digit:
    mov cl, [si]
    cmp cl, 0
    je .finished
```

```
    sub cl, '0'
    mov dx, 10
    mul dx              ; AX = AX * 10
    add ax, cx
    inc si
    jmp .next_digit
.finished:
    ret
```

RUN OUTPUT — assembled program, 16-bit emulator

AFD debugger is not available in this environment, so the program was run in a 16-bit emulator instead. AFD screenshot must be captured by the student with their own VUID before submitting.

```
VUID in memory :    BC123456789
prefix         :    BC
LHS (text)     :    1234        LHS_VAL : 1234  (04D2h)
RHS (text)     :    56789       RHS_VAL : 56789 (DDD5h)

GCD divisions  :    56789 / 1234 -> rem 25
                    1234 / 25 -> rem 9
                    25 / 9 -> rem 7
                    9 / 7 -> rem 2
                    7 / 2 -> rem 1
                    2 / 1 -> rem 0

BX (GCD)       :    0001h
```