CPSC 240: Computer Organization and Assembly Language
Assignment 04, Fall Semester 2023

CWID:____________________ Name:____________________________

1. Download the “CPSC-240 Assignment04.docx” document.
2. Design the "multiple.asm" program, and use assembly language to realize the function of the following C++ instructions.

   unsigned short num = 225;

   unsigned short mul_15 = 0, other = 0;

   if(num % 3 == 0 && num % 5 == 0) {

       mul_15++;

   } else {

       other++;

   }
3. Assemble the "multiple.asm" file and link the "multiple.o" file to get the "multiple" executable file.
4. Run the "multiple" file with the DDD debugger to display the memory of num, as well as the simulation results of mul15 and other.
5. Insert source code (multiple.asm) and simulation results (GDB window) of the memory (num, mul_15, and other) in the document. Write an analysis to verify simulation results.
6. Save the file in pdf format and submit the pdf file to Canvas before 23:59 pm on 10/05/2023.

[Insert multiple.asm source code here]

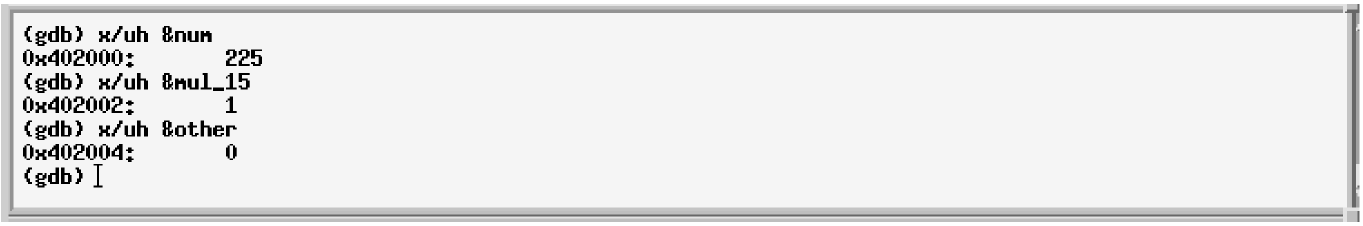
```
; multiple.asm
; unsigned short num = 225;
; unsigned short mul_15 = 0, other = 0;
; if(num % 3 == 0 && num % 5 == 0) {
;    mul_15++;
; } else {
;    other++;
; }


section .data
    num     dw      225
    mul_15  dw      0                        ;mul_15 = 0
    other   dw      0                        ;other = 0

section .text
    global _start
_start:
    mov     ax, word[num]                    ;ax = num
    mov     bl, 3                            ;bl = 3
    div     bl                               ;ah = ax%bl, al = ax/bl
    cmp     ah, 0                            ;compare ah,0
    jne     else                             ;if(ax%3!=0) goto else
    mov     ax, word[num]                    ;ax = num
    mov     bl, 5                            ;bl = 5
    div     bl                               ;ah = ax%bl, al = ax/bl
    cmp     ah, 0                            ;compare ah,0
```

```
     jne     else                                  ;if(ax%5!=0) goto else
     inc     word[mul_15]                          ;mul_15 = mul_15 + 1
     jmp     end_if                                ;goto end_if
else:
     inc     word[other]                           ;other = other + 1
end_if:
     mov     rax, 60                               ;terminate excuting process
     mov     rdi, 0                                ;exit status
     syscall                                       ;calling system services
```

[Insert multiple simulation result here]



```
(gdb) x/uh &num
0x402000:       225
(gdb) x/uh &mul_15
0x402002:       1
(gdb) x/uh &other
0x402004:       0
(gdb) 
```

[Insert multiple simulation result verification here]

225 % 3 = 0

225 % 5 = 0

or

225 mod 3 = 0

225 mod 5 = 0