CPSC 240: Computer Organization and Assembly Language
Assignment 03, Fall Semester 2023

CWID:__________________ Name:__________________________

1. Download the “CPSC-240 Assignment03.docx” document.
2. Design the "multiplication.asm" program, and use assembly language to realize the function of the following C++ instructions.

   unsigned int num1 = 300,000;
   unsigned int num2 = 400,000;
   unsigned long product = 0;
   product = long(num1 * num2);
3. Assemble the "multiplication.asm" file and link the "multiplication.o" file to get the "multiplication" executable file.
4. Run the "multiplication" file with the DDD debugger to display the simulation results of num1 and num2, as well as the simulation results of product.
5. Insert source code (multiplication.asm) and simulation results (GDB panel) of the memory (num1, num2, and product) in the document. Use calculator or hand calculation to verify simulation results.
6. Design the "division.asm" program, and use assembly language to realize the function of the following C++ instructions.

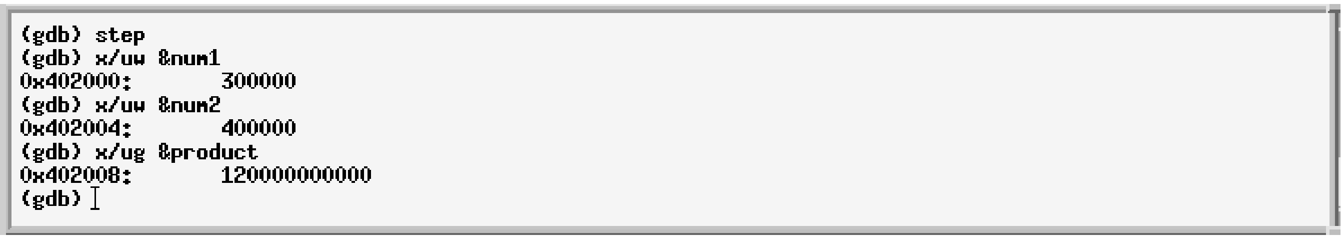
   unsigned long num1 = 50,000,000,000;
   unsigned int num2 = 3,333,333;
   unsigned int quotient = 0, remainder = 0;
   quotient = num1 / num2;
   remainder = num1 % num2;
7. Assemble the "division.asm" file and link the "division.o" file to get the "division" executable file.
8. Run the "division" file with the DDD debugger to display the simulation results of num1 and num2, as well as the simulation results of quotient and remainder.
9. Insert source code (division.asm) and simulation results (GDB panel) of the memory (num1, num2, quotient, and remainder) in the document. Use calculator or hand calculation to verify simulation results.
10. Save the file in pdf format and submit the pdf file to Canvas before 23:59 pm on 09/20/2023.

[Insert multiplication assembly source code here]

```
;multiplication.asm
;unsigned int num1 = 300,000;
;unsigned int num2 = 400,000;
;unsigned long product = 0;
;product = long(num1 * num2);

section .data
        num1    dd      300000                  ;num1 = 0004 93E0h
        num2    dd      400000                  ;num2 = 0006 1A80h
        product dq      0                       ;product = 0000 0000 0000 0000h

section .text
```

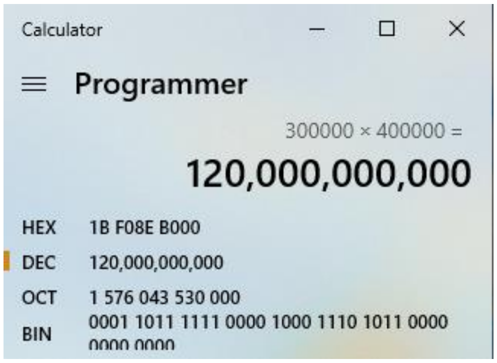
```
        global _start

_start:
        Mov     eax, dword[num1]                ;eax = num1 = 0004 93E0h
        Mul     dword[num2]                 ;edx:eax = eax * num2 = 0000 001B F08E B000h
        Mov     dword[product], eax             ;product+0 = eax = F08E B000h
        Mov     dword[product+4], edx           ;product+8 = eax = 0000 001Bh

        Mov     rax, 60                         ;terminate excuting process
        Mov     rdi, 0                          ;exit status
        syscall                                 ;calling system services
```

[Insert multiplication simulation results (GDB panel) here]



```
(gdb) step
(gdb) x/uw &num1
0x402000:       300000
(gdb) x/uw &num2
0x402004:       400000
(gdb) x/ug &product
0x402008:       120000000000
(gdb) 
```

[Insert verification of calculator result or hand calculation here]



300000 × 400000 =

120,000,000,000

HEX 1B F08E B000

DEC 120,000,000,000

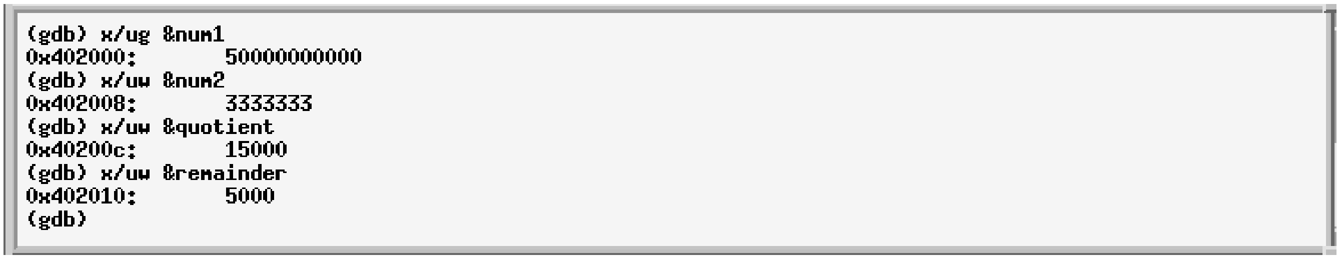
OCT 1 576 043 530 000

BIN 0001 1011 1111 0000 1000 1110 1011 0000
0000 0000

[Insert division assembly source code here]

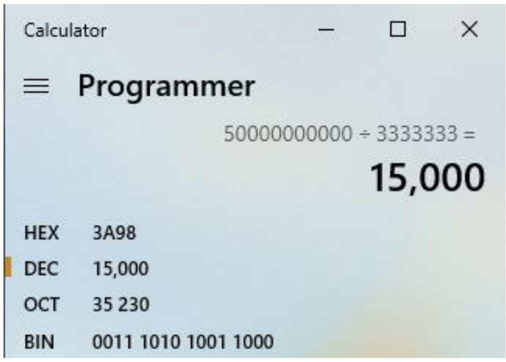
```
;division.asm
;unsigned long num1 = 50,000,000,000;
;unsigned int num2 = 3,333,333;
;unsigned int quotient = 0, remainder = 0;
;quotient = num1 / num2;
;remainder = num1 % num2;

section .data
    num1            dq      50000000000         ;num1 = 0000 000B A43B 7400h
    num2            dd      3333333             ;num2 = 0032 DCD5h
    quotient        dd      0                   ;quotient = 0000 0000h
    remainder       dd      0                   ;remainder = 0000 0000h

section .text
    global _start

_start:
    mov     edx, dword[num1+4]                  ;edx = num1+4 = A43B 7400h
    mov     eax, dword[num1+0]                  ;eax = num1+0 = 0000 000Bh
    div     dword[num2]                         ;eax=edx:eax/num2=3A98h=15000
                                                ;edx=edx:eax%num2=1388h=5000
    Mov     dword[quotient], eax                ;quotient = eax = 3A98h = 15000
    Mov     dword[remainder], edx            ;remainder = edx = 1388h = 5000

    Mov     rax, 60                          ;terminate excuting process
    Mov     rdi, 0                           ;exit status
    syscall                                  ;calling system services
```

[Insert division simulation results (GDB panel) here]



```
(gdb) x/ug &num1
0x402000:       50000000000
(gdb) x/uw &num2
0x402008:       3333333
(gdb) x/uw &quotient
0x40200c:       15000
(gdb) x/uw &remainder
0x402010:       5000
(gdb)
```

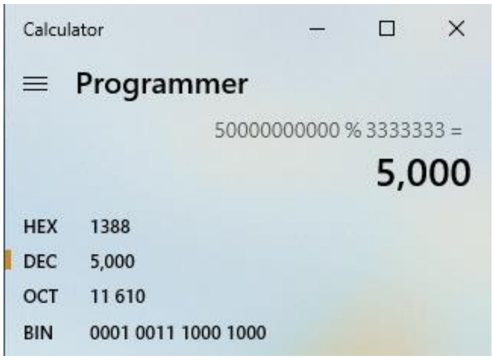
[Insert verification of calculator result or hand calculation here]



50000000000 ÷ 3333333 =

15,000

```
HEX   3A98
DEC   15,000
OCT   35 230
BIN   0011 1010 1001 1000
```



50000000000 % 3333333 =

5,000

```
HEX   1388
DEC   5,000
OCT   11 610
BIN   0001 0011 1000 1000
```