Computer Science Department
California State University, Fullerton

CPSC 240 Computer Organization and Assembly Language
Quiz 02
12:00 PM to 1:15 PM
Thursday, March 23, 2023

Student Name:

Last 4 digits of ID:

Note:

- University regulations on academic honesty will be strictly enforced.
- You have 75 minutes to complete this Quiz.
- Open books, slides and sample programs.
- Turn off or turn vibration your cell phone.
- Use YASM assembler for the program design.
- Copy and paste your assembly source code and DDD debugger window to the end of the word file and save it in pdf or docx format.
- Submit you pdf or docx file to Canvas before the deadline.
  NOTE: Email submissions will not be graded.
- Any content submitted after the due date will be regarded as a make-up quiz.

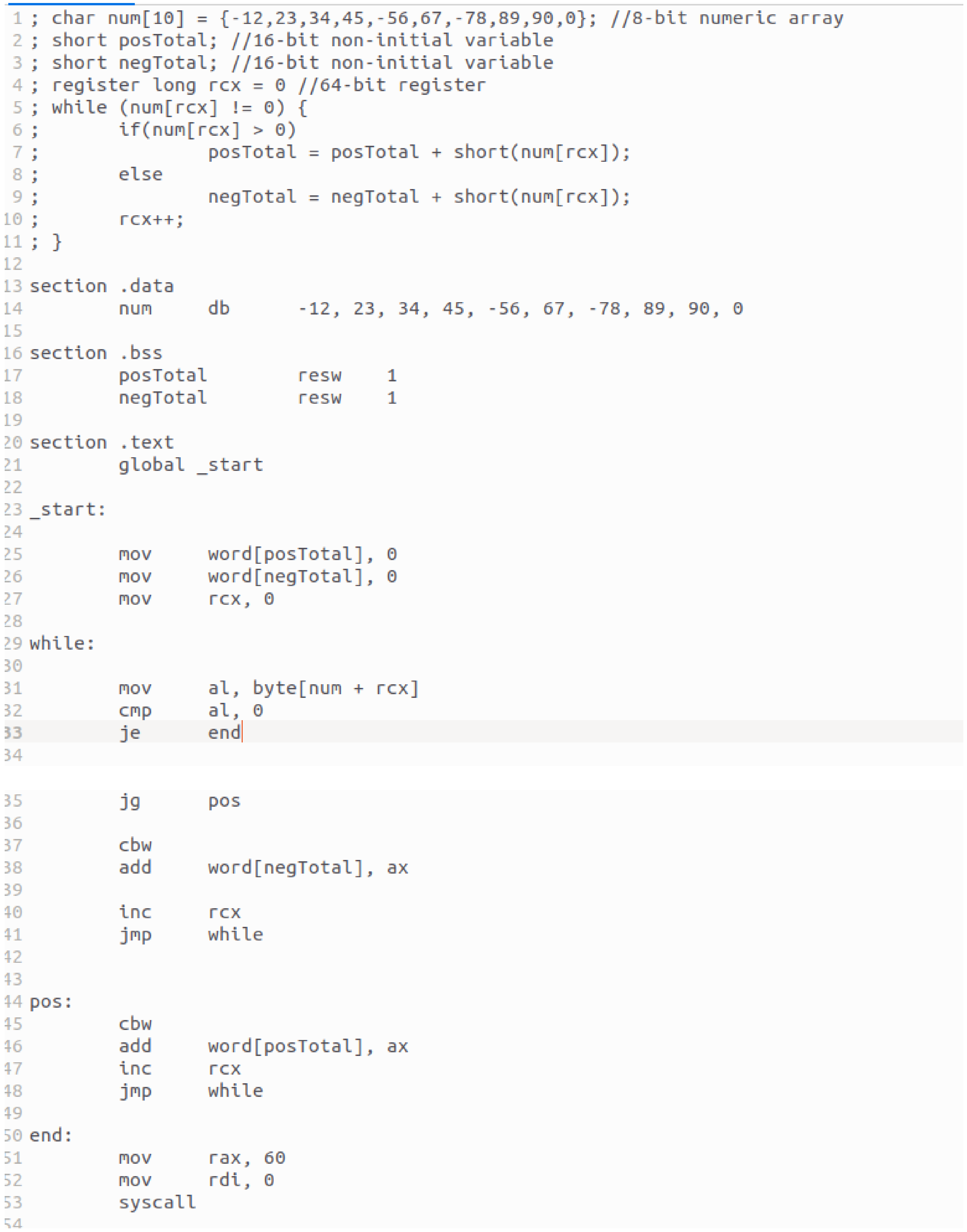
Quiz 02

1. Download the "CPSC-240 Spring 2023 Quiz 02_1.docx" document.
2. Use x86-64 assembly language to implement the following C/C++ arithmetic operations.

```
char num[10] = {-12,23,34,45,-56,67,-78,89,90,0};      //8-bit numeric array
short posTotal;                                        //16-bit non-initial variable
short negTotal;                                        //16-bit non-initial variable
register long rcx = 0                                  //64-bit register
while (num[rcx] != 0) {
    if(num[rcx] > 0)
        posTotal = posTotal + short(num[rcx]);
    else
        negTotal = negTotal + short(num[rcx]);
    rcx++;
}
```

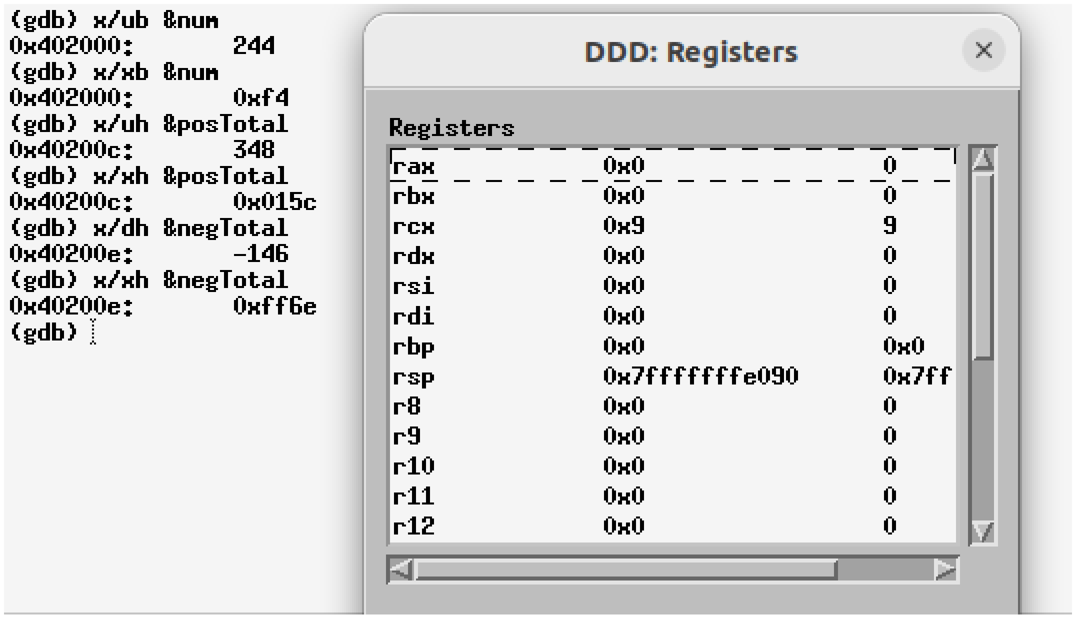
3. After assembling and linking, run the DDD debugger to display the simulation results of the register window before terminate program and the memories of num, posTotal, and negTotal.
4. Insert source code and the simulation results (GDB window) to the bottom of the document.
5. Save the file in pdf or docx format and submit the pdf or docx file to Canvas before the deadline.
6. Deadline is 1:15 pm on 03/23/2023.

[Attach your assembly source code here:]



```
; char num[10] = {-12,23,34,45,-56,67,-78,89,90,0}; //8-bit numeric array
; short posTotal; //16-bit non-initial variable
; short negTotal; //16-bit non-initial variable
; register long rcx = 0 //64-bit register
; while (num[rcx] != 0) {
;       if(num[rcx] > 0)
;               posTotal = posTotal + short(num[rcx]);
;       else
;               negTotal = negTotal + short(num[rcx]);
;       rcx++;
; }

section .data
        num     db      -12, 23, 34, 45, -56, 67, -78, 89, 90, 0

section .bss
        posTotal        resw    1
        negTotal        resw    1

section .text
        global _start

_start:

        mov     word[posTotal], 0
        mov     word[negTotal], 0
        mov     rcx, 0

while:

        mov     al, byte[num + rcx]
        cmp     al, 0
        je      end


        jg      pos

        cbw
        add     word[negTotal], ax

        inc     rcx
        jmp     while


pos:
        cbw
        add     word[posTotal], ax
        inc     rcx
        jmp     while

end:
        mov     rax, 60
        mov     rdi, 0
        syscall

```

[Attach GDB window with all memory data here:]



```
(gdb) x/ub &num
0x402000:       244
(gdb) x/xb &num
0x402000:       0xf4
(gdb) x/uh &posTotal
0x40200c:       348
(gdb) x/xh &posTotal
0x40200c:       0x015c
(gdb) x/dh &negTotal
0x40200e:       -146
(gdb) x/xh &negTotal
0x40200e:       0xff6e
(gdb)
```

```
Registers
rax            0x0               0
rbx            0x0               0
rcx            0x9               9
rdx            0x0               0
rsi            0x0               0
rdi            0x0               0
rbp            0x0               0x0
rsp            0x7fffffffe090    0x7ff
r8             0x0               0
r9             0x0               0
r10            0x0               0
r11            0x0               0
r12            0x0               0
```