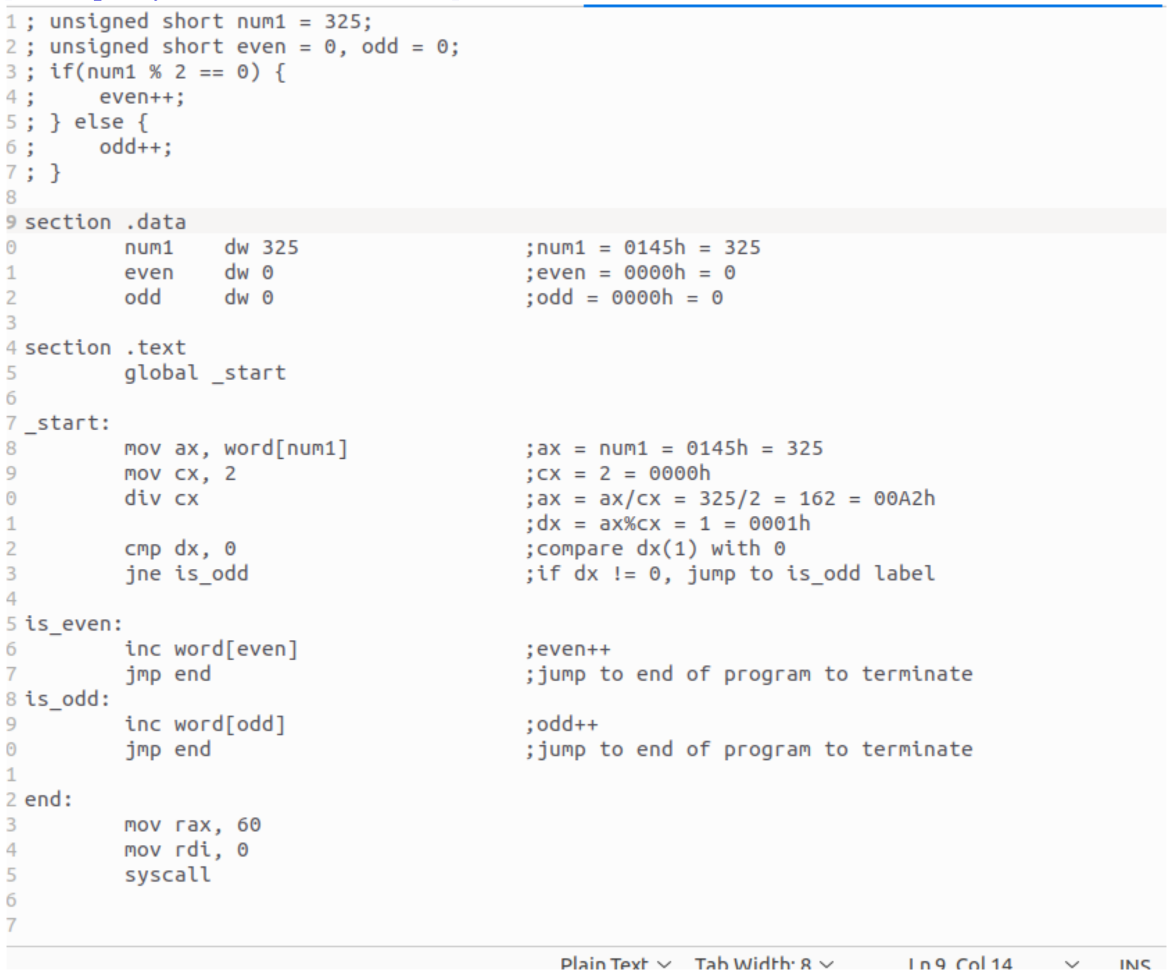
CPSC 240: Computer Organization and Assembly Language
Assignment 04, Spring Semester 2023

CWID: 886880327   Name: Andrew Saldana

1. Download the "CPSC-240 Assignment04.docx" document.
2. Design the "parity.asm" program, and use assembly language to realize the function of the following C++ instructions.
   unsigned short num1 = 325;
   unsigned short even = 0, odd = 0;
   if(num1 % 2 == 0) {
       even++;
   } else {
       odd++;
   }
3. Assemble the " parity.asm" file and link the "parity.o" file to get the " parity" executable file.
4. Run the "parity" file with the DDD debugger to display the memory of num1, as well as the simulation results of even and odd.
5. Insert source code (parity.asm) and simulation results (GDB window) of the memory (num1, even, and odd) 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 03/05/2023.

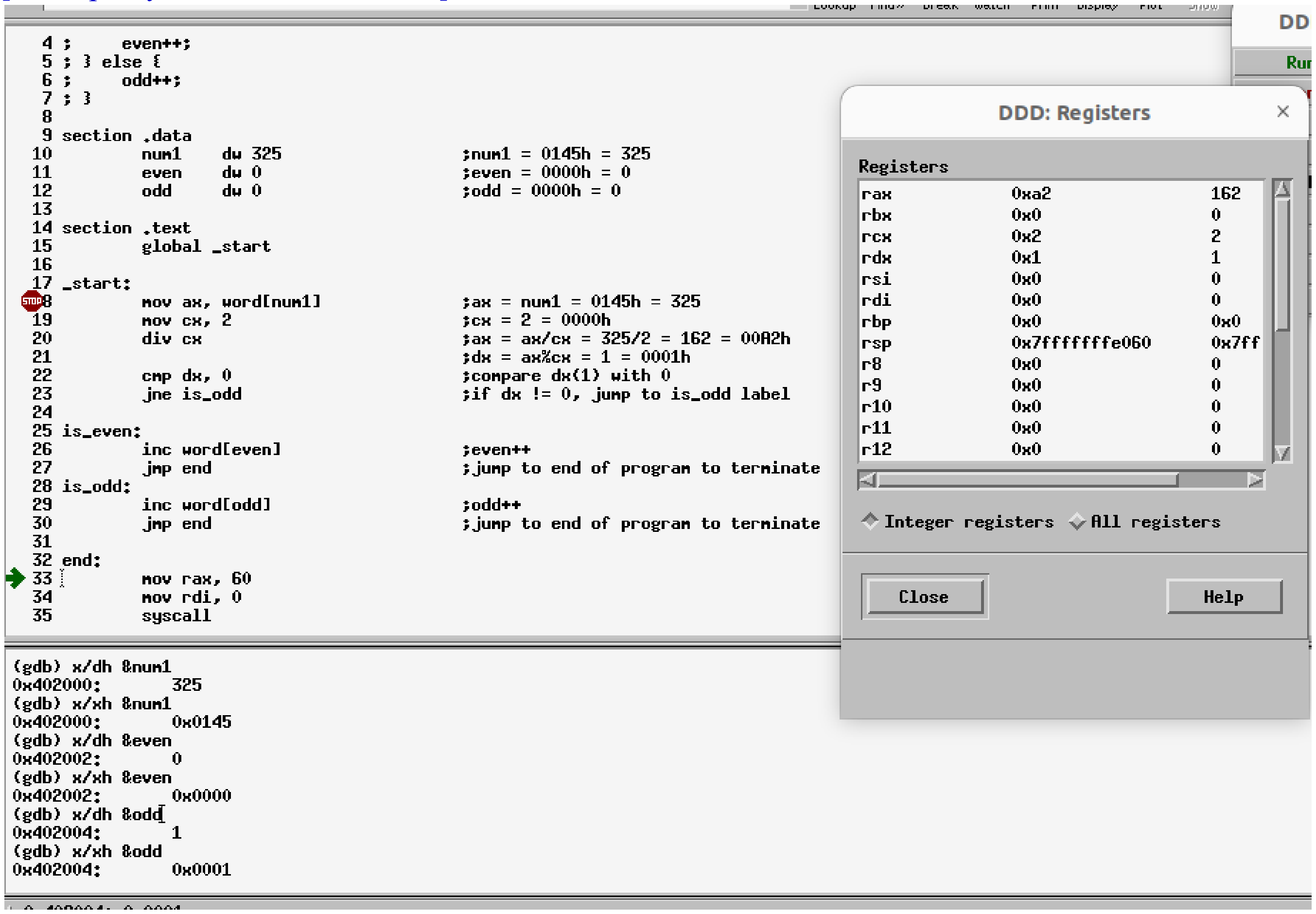
[Insert parity.asm source code here]



```
; unsigned short num1 = 325;
; unsigned short even = 0, odd = 0;
; if(num1 % 2 == 0) {
;     even++;
; } else {
;     odd++;
; }

section .data
        num1    dw 325                  ;num1 = 0145h = 325
        even    dw 0                    ;even = 0000h = 0
        odd     dw 0                    ;odd = 0000h = 0

section .text
        global _start

_start:
        mov ax, word[num1]              ;ax = num1 = 0145h = 325
        mov cx, 2                       ;cx = 2 = 0000h
        div cx                          ;ax = ax/cx = 325/2 = 162 = 00A2h
                                        ;dx = ax%cx = 1 = 0001h
        cmp dx, 0                       ;compare dx(1) with 0
        jne is_odd                      ;if dx != 0, jump to is_odd label

is_even:
        inc word[even]                  ;even++
        jmp end                         ;jump to end of program to terminate
is_odd:
        inc word[odd]                   ;odd++
        jmp end                         ;jump to end of program to terminate

end:
        mov rax, 60
        mov rdi, 0
        syscall
```

Plain Text ∨   Tab Width: 8 ∨   Ln 9, Col 14   ∨   INS

[Insert parity simulation result here]



```
;       even++;
; } else {
;       odd++;
; }

section .data
        num1    dw 325                  ;num1 = 0145h = 325
        even    dw 0                    ;even = 0000h = 0
        odd     dw 0                    ;odd = 0000h = 0

section .text
        global _start

_start:
        mov ax, word[num1]              ;ax = num1 = 0145h = 325
        mov cx, 2                       ;cx = 2 = 0000h
        div cx                          ;ax = ax/cx = 325/2 = 162 = 00A2h
                                        ;dx = ax%cx = 1 = 0001h
        cmp dx, 0                       ;compare dx(1) with 0
        jne is_odd                      ;if dx != 0, jump to is_odd label

is_even:
        inc word[even]                  ;even++
        jmp end                         ;jump to end of program to terminate
is_odd:
        inc word[odd]                   ;odd++
        jmp end                         ;jump to end of program to terminate

end:
        mov rax, 60
        mov rdi, 0
        syscall
```

```
(gdb) x/dh &num1
0x402000:       325
(gdb) x/xh &num1
0x402000:       0x0145
(gdb) x/dh &even
0x402002:       0
(gdb) x/xh &even
0x402002:       0x0000
(gdb) x/dh &odd
0x402004:       1
(gdb) x/xh &odd
0x402004:       0x0001
```

DDD: Registers

Registers

```
rax     0xa2                162
rbx     0x0                 0
rcx     0x2                 2
rdx     0x1                 1
rsi     0x0                 0
rdi     0x0                 0
rbp     0x0                 0x0
rsp     0x7fffffffe060      0x7ff
r8      0x0                 0
r9      0x0                 0
r10     0x0                 0
r11     0x0                 0
r12     0x0                 0
```

[Insert parity simulation result analysis here]

325 in hex

$$\left.\begin{array}{llll} 325/16 & = & 20 & R\ 5 \\ 20/16 & = & 1 & R\ 4 \\ 1/16 & = & 0 & R\ 1 \end{array}\right\}$$ hex (16 bits) 0145h

---

325/2 = 162 R 1

162 in hex

$$\left.\begin{array}{llll} 162/16 & = & 10 & R\ 2 \\ 10/16 & = & 0 & R\ 10 \end{array}\right\}$$ hex in 16 bits 00A2h

		362/2		compare dx(1) and 0 jne is_odd		
num	divisor	ax	dx	jump?	even	odd
325	2	162	1	yes, jump to is_odd:	0	odd++ 1

odd = 1
even = 0