CPSC 240: Computer Organization and Assembly Language
Assignment 05, Fall Semester 2023

CWID:_____ Name:_____

1. Download the "CPSC-240 Assignment05.docx" document.
2. Convert the following C/C++ variable declarations and arithmetic operations to x86-64 assembly language. Use the "yasm" assembler to assemble the program, the "ld" linker to link the object code, and the "ddd" debugger to simulate the executable code.
   NOTE: variable sizes and program functions should be equivalent to C/C++ instructions.

```
unsigned short array[7] = {12, 1003, 6543, 24680, 789, 30123, 32766};
unsigned short even[7];
register long rsi = 0, rdi = 0;
do {
    if(array[rsi] % 2 == 0) {
        even[rdi] = array[rsi];
        rdi++;
    }
    rsi++;
} while(rsi < 7);
```

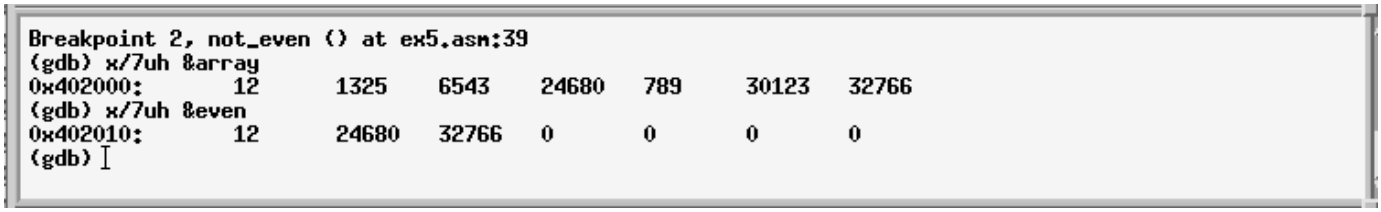
3. Assemble the "doWhile.asm" file and link the "parity.o" file to get the "parity" executable file.
4. Run the "parity" file with the DDD debugger to display the simulation results of array and even.
5. Insert source code (parity.asm) and simulation results (GDB window) of the memory array (array and even) in the document. Use hand calculation to verify simulation results.
6. Save the file in pdf or docx format and submit the pdf or docx file to Canvas before 23:59 pm on 10/12/2023.

[Insert the source code of parity.asm here]

```
;parity.asm
;unsigned short array[7] = {12, 1003, 6543, 24680, 789, 30123, 32766};
;unsigned short even[7];
;register long rsi = 0, rdi = 0;
;do {
;    if(array[rsi] % 2 == 0) {
;            even[rdi] = array[rsi];
;            rdi++;
;    }
;    rsi++;
;} while(rsi < 7);

section .data
array       dw      12, 1325, 6543, 24680, 789, 30123, 32766

section .bss
even        resw    7

section .text
```

```
    global _start
_start:
    mov     rsi, 0                          ;rsi = 0
    mov     rdi, 0                          ;rdi = 0
    mov     bx, 2                           ;bx = 2
doloop:
    mov     ax, word[array+(rsi*2)]         ;ax = array[rsi]
    cwd                                     ;convert ax to dx:ax
    div     bx                              ;dx = dx:ax % bx
    cmp     dx, 0                           ;compare dx and 0
    jne     not_even                        ;if(remainder==0) {
    mov     r8w, word[array+(rsi*2)]        ;    r8w = array[rsi]
    mov     word[even+(rdi*2)], r8w         ;    even[rdi] = r8w
    inc     rdi                             ;rdi = rdi + 1
not_even:                                   ;{
    inc     rsi                             ;rsi = rsi + 1
    cmp     rsi, 7                          ;compare rsi and 7
    jb      doloop                          ;if(rsi<7) goto doloop

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

[Insert parity simulation result (GDB window with array and even) here]



[Insert verification of hand calculation here]

12 % 2 = 0

1325 % 2 = 1

6543 % 2 = 1

24680 % 2 = 0

789 % 2 = 1

30123 % 2 = 1

32766 % 2 = 0