CPSC 240: Computer Organization and Assembly Language
Assignment 07, Fall Semester 2023

CWID:_885857847_____ Name:Kush Patel_____

1. Download the "CPSC-240 Assignment07.docx" document.
2. Design the "input.asm" program, input 9 values from 1 to 9 from the keyboard, find out the multiples of 3 from the input values, and display the multiples of 3 in the terminal emulator window. The corresponding C/C++ code is as follows:

```
char msg1[] = "Input a number (1~9): ";
char msg2[] = " is Multiple of 3.";
char buffer;
char num;
char ascii[10];

register int r10 = 0;
do {
    cout << msg1;
    cin >> buffer;
    ascii[r10] = buffer;
    r10++;
} while(r10 < 9);
r10 = 0;
do {
    num = atoi(ascii[r10]);
    if(num%3 == 0) {
        cout << ascii[r10] << msg2;
    }
    r10++;
} while(r10 < 9);
```

3. Assemble the "input.asm" file and link the "input.o" file to get the "input" executable file.
4. Run the "input" file to display the input value and multiple of 3 in Terminal Emulator window.
5. Insert source code (input.asm) and simulation results (Terminal Emulator window) at the bottom of the document. Write an analysis to verify the simulation results.
6. Save the file in pdf format and submit the pdf file to Canvas before 23:59 pm on 11/02/2023.

Sample output:

Alternatively, the corresponding C/C++ code can be replaced as follows:

```
char num;
char buffer;
char msg1[] = "Input a number (1~9): ";
char msg2[] = " is multiple of 3.";

register int r10 = 0;
do {
    cout << msg1;
    cin >> buffer;
    num = atoi(buffer);
    if(num%3 == 0) {
        cout << buffer << msg2;
    }
    r10++;
} while(r10 < 9);
```

Sample output:

[Insert input.asm here]

```
1
2  section .bss
3          num     resb    1                                       ;reserve 1-byte for num
4          buffer  resb    2                                       ;reserve 1-byte for buffer
5
6  section .data
7          mesg    db      "Input a number (0~9): "                ;input message
8          mul_3    db      " is an multiple of 3", 10               ;even message
9
10
11 section .text
12         global _start
13 _start:
14         mov     r10, 0
15 a_loop:
16         ; cout << mesg
17         mov     rax, 1                                          ;SYS_write
18         mov     rdi, 1                                          ;write to STD_OUT
19         mov     rsi, mesg                                       ;address of mesg
20         mov     rdx, 22                                         ;22 character to write
21         syscall                                                 ;calling system services
22
23         ; cin >> num
24         mov     rax, 0                                          ;SYS_read
25         mov     rdi, 0                                          ;read from STD_IN
26         mov     rsi, buffer                                     ;address of the buffer
27         mov     rdx, 2                                          ;input length = 1
28         syscall                                                 ;calling system services
29         mov     al, byte[buffer]                                ;al = buffer (ex: '5'=35h)
30         and     al, 0fh                                         ;al = block bit7~4 (ex: 05h)
31         mov     byte[num], al                                   ;num = al (ex: num=05h)
32
33         ; if(num%2 == 0)
34         mov     ah,0                                            ;ah=0
35         mov     al, byte[num]                                   ;al=num
36         mov     bl, 3                                           ;bl=2
37         div     bl                                              ;ah=ax%bl, al=ax/bl
38         cmp     ah, 0                                           ;compare ah,0
39         jnz     done                                           ;if(rem!=0) goto odd_num
40         ; cout << num
41         mov     rax, 1                                          ;SYS_write
42         mov     rdi, 1                                          ;where to write
43         mov     rsi, buffer                                     ;address of buffer
44         mov     rdx, 1                                          ;1 character to write
45         syscall                                                 ;calling system services
46         ; cout << even
47         mov     rax, 1                                          ;SYS_write
```

```
16          ; cout << mesg
17          mov     rax, 1                          ;SYS_write
18          mov     rdi, 1                          ;write to STD_OUT
19          mov     rsi, mesg                       ;address of mesg
20          mov     rdx, 22                         ;22 character to write
21          syscall                                 ;calling system services
22
23          ; cin >> num
24          mov     rax, 0                          ;SYS_read
25          mov     rdi, 0                          ;read from STD_IN
26          mov     rsi, buffer                     ;address of the buffer
27          mov     rdx, 2                          ;input length = 1
28          syscall                                 ;calling system services
29          mov     al, byte[buffer]                ;al = buffer (ex: '5'=35h)
30          and     al, 0fh                         ;al = block bit7~4 (ex: 05h)
31          mov     byte[num], al                   ;num = al (ex: num=05h)
32
33          ; if(num%2 == 0)
34          mov     ah,0                            ;ah=0
35          mov     al, byte[num]                   ;al=num
36          mov     bl, 3                           ;bl=2
37          div     bl                              ;ah=ax%bl, al=ax/bl
38          cmp     ah, 0                           ;compare ah,0
39          jnz     done                          ;if(rem!=0) goto odd_num
40          ; cout << num
41          mov     rax, 1                          ;SYS_write
42          mov     rdi, 1                          ;where to write
43          mov     rsi, buffer                     ;address of buffer
44          mov     rdx, 1                          ;1 character to write
45          syscall                                 ;calling system services
46          ; cout << even
47          mov     rax, 1                          ;SYS_write
48          mov     rdi, 1                          ;where to write
49          mov     rsi, mul_3                      ;address of num
50          mov     rdx, 22                         ;18 character to write
51          syscall                                 ;calling system services
52          jmp     done                            ;jump to done
53          ; else
54
55  done:
56          inc    r10
57          cmp r10, 9
58          jl a_loop
59          mov      rax, 60                         ;terminate excuting process
60          mov     rdi, 0                          ;exit status
61          syscall                                 ;calling system services
```

[Insert input simulation result here]

```
Warning: Cannot convert string "-*-symbol-*-*-*-*-*-120-*-*-*-*-adobe-*" to type
 FontStruct
Warning: Could not load font "-*-lucidatypewriter-medium-*-*-*-*-120-*-*-*-*-iso
8859-*", using font "fixed" instead
Warning: Could not load font "-*-lucidatypewriter-medium-r-*-*-*-120-*-*-*-*-iso
8859-*", using font "fixed" instead
Warning: Could not load font "-*-lucidatypewriter-bold-*-*-*-*-120-*-*-*-*-iso88
59-*", using font "fixed" instead
student@tuffix-vm:~/Desktop/input$ ./input
Input a number (0~9): 0
Visual Studio Code
of 3
~9): 1
Input a number (0~9): 2
Input a number (0~9): 3
3 is an multiple of 3
Input a number (0~9): 4
Input a number (0~9): 5
Input a number (0~9): 6
6 is an multiple of 3
Input a number (0~9): 7
Input a number (0~9): 8
student@tuffix-vm:~/Desktop/input$ 9
9: command not found
student@tuffix-vm:~/Desktop/input$
```

```
student@tuffix-vm:~$ cd Desktop
student@tuffix-vm:~/Desktop$ cd input
student@tuffix-vm:~/Desktop/input$ ./input
Input a number (0~9): 9
9 is an multiple of 3
Input a number (0~9): 10
Input a number (0~9): Input a number (0~9): 12
Input a number (0~9): Input a number (0~9): 16
Input a number (0~9): Input a number (0~9): 16
Input a number (0~9): student@tuffix-vm:~/Desktop/input$
4: command not found
student@tuffix-vm:~/Desktop/input$
```

[Insert input simulation result verification here]

```
{
char num;
char buffer;
char msg1[] = "Input a number (1~9): ";
char msg2[] = " is multiple of 3.";

register int r10 = 0;
do {
    cout << msg1;
    cin >> buffer;
    num = buffer;
    if(num%3 == 0) {
        cout << buffer << msg2;
    }
    r10++;
} while(r10 < 9);

    return 0;
}

```

```
nput a number (1~9): 1
nput a number (1~9): 2
nput a number (1~9): 3
 is multiple of 3.Input a number (1~9): 4
nput a number (1~9): 5
nput a number (1~9): 6
```

```
19  char msg2[] = " is multiple of 3.";
20
21  register int r10 = 0;
22  do {
23      cout << msg1;
24      cin >> buffer;
25      num = buffer;
26      if(num%3 == 0) {
27          cout << buffer << msg2;
28      }
29      r10++;
30  } while(r10 < 9);
31
32      return 0;
33  }
34
```

```
Input a number (1~9): 6
6 is multiple of 3.Input a number (1~9): 7
Input a number (1~9): 8
Input a number (1~9): 9
9 is multiple of 3.
```