CSD 1100 ASM Practice (Quiz + Slides)

By Huang Wei Jhin 2021

Question **1**

Correct

1.00 points out of 1.00

What is the output of the following code?

```
4   section .data
5   fmt     db "%lld",10,0
6   n       dq 10
7
8   section .text
9       global _start
10      extern printf
11
12  _start:
13      mov rdi, 20
14      mov rsi, rdi
15      mov [n], rsi
16
17      mov rdi, fmt
18      mov rsi, [n]
19      xor rax, rax
20      call printf
21
22      mov rax, 60
23      mov rdi, 0
24      syscall
```

Select one:

- 0
- 10
- 20
- 60
- Code is not compilable

Question **2**

What is the output of the following code?

Correct

1.00 points out of 1.00

```
3   section .data
4   fmt db '%lld',10,0
5
6   section .text
7       global _start
8       extern printf
9
10  _start:
11      push 10
12      push 20
13
14      mov rdi, fmt
15      pop rsi
16      pop r15
17      xor rax, rax
18      call printf
19
20      mov rax, 60
21      mov rdi, 0
22      syscall
```

Select one:

- 0
- 10
- 20
- 60
- Code is not compilable

Question **3**

Incorrect

0.00 points out of 1.00

What is the output of the following code?

```
4   section .data
5   fmt db '%lld',10,0
6
7   section .text
8       global _start
9       extern printf
10
11  _start:
12      mov rdx, 0
13      mov rax, 5
14      mov r15, 10
15      idiv r15
16
17      mov rdi, fmt
18      mov rsi, rdx
19      xor rax, rax
20      call printf
21
22      mov rax, 60
23      mov rdi, 0
24      syscall
```

Select one:

- 0
- 5
- 10
- 60
-

Code is not compilable

Question **4**

Correct

1.00 points out of 1.00

What is the output of the following code?

```
section .data
fmt db '%lld',10,0

section .text
    global _start
    extern printf

_start:
    mov rdi, 10
    xor rsi, rsi
    add rsi, rdi

    mov rdi, fmt
    xor rax, rax
    call printf

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

Select one:

- 0
- 10
- 20
- 60
- Code is not compilable

Started on	Wednesday, November 17, 2021, 11:06 AM
State	Finished
Completed on	Wednesday, November 17, 2021, 11:18 AM
Time taken	11 mins 45 secs
Points	3.00/4.00
Grade	**75.00** out of 100.00

Question **1**

Correct

1.00 points out of 1.00

What is the output of the following code?

```
section .data
fmt db '%lld',10,0

section .text
    global _start
    extern printf

_start:
    push 10
    push 20
    push 30

    mov rdi, fmt
    pop rsi
    pop rsi
    pop r15
    xor rax, rax
    call printf

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

Select one:

- 0
- 10
- 20
- 30
- Code is not compilable

Question **2**

What is the output of the following code?

Incorrect

0.00 points out of 1.00

```
6   section .data
7   fmt     db "%lld",10,0
8   n       dq 10
9   m       dq 0
10
11  section .text
12      global _start
13      extern printf
14
15  _start:
16      mov [n], [m]
17
18      mov rdi, fmt
19      mov rsi, [n]
20      xor rax, rax
21      call printf
22
23      mov rax, 60     ; syscall number for exit
24      mov rdi, 0      ; int status 0
25      syscall
```

Select one:

- 0
- 10
- 20
- 60
- Code is not compilable

Question **3**

Correct

1.00 points out of 1.00

What is the output of the following code?

```
6   section .data
7   fmt  db '%lld',10,0
8
9   section .text
10      global _start
11      extern printf
12
13  _start:
14      mov rdi, 10
15      xor rsi, rsi
16      sub rsi, rdi
17
18      mov rdi, fmt
19      xor rax, rax
20      call printf
21
22      mov rax, 60
23      mov rdi, 0
24      syscall
```

Select one:

- 0
- 10
- -10
- 60
- Code is not compilable

Question **4**

Correct

1.00 points out of 1.00

What is the output of the following code?

```
4   section .data
5   fmt db '%lld',10,0
6
7   section .text
8       global _start
9       extern printf
10
11  _start:
12      mov rdx, 0
13      mov rax, 15
14      mov r15, 10
15      idiv r15
16
17      mov rdi, fmt
18      mov rsi, rax
19      xor rax, rax
20      call printf
21
22      mov rax, 60
23      mov rdi, 0
24      syscall
```

Select one:

- 0
- 1
- 5
- 10
- Code is not compilable

Started on	Monday, November 22, 2021, 11:09 AM
State	Finished
Completed on	Monday, November 22, 2021, 11:20 AM
Time taken	10 mins 23 secs
Points	4.00/5.00
Grade	**80.00** out of 100.00

Question **1**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
27    section .data
28    fmt     db "%lld",10,0
29
30    section .text
31        global _start
32        extern printf
33
34    _start:
35        PRINTF fmt, 10
36        jmp next
37        PRINTF fmt, 20
38    next:
39        PRINTF fmt, 30
40        EXIT 0
```

Select one:

- Outputs nothing
- Outputs 10
- Outputs 10 and 20
- Outputs 10 and 30
- Outputs 20 and 30
- Outputs 10, 20 and 30

Question **2**

Incorrect

0.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
27    section .data
28    fmt     db "%lld",10,0
29
30    section .text
31        global _start
32        extern printf
33
34    _start:
35        mov rbx, 20
36        cmp rbx, 10
37        je next
38        PRINTF fmt, 10
39    next:
40        PRINTF fmt, 20
41        EXIT 0
```

Select one:

- Outputs nothing
- Outputs 10
- Outputs 10 and 20
- Outputs 10 and 30
- Outputs 20 and 30
- Outputs 10 20 and 30

Question **3**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
26  section .data
27  fmt     db "%lld",10,0
28  a       dq 20
29  b       dq 10
30
31  section .text
32      global _start
33      extern printf
34
35  _start:
36      mov rbx, [b]
37      cmp rbx, [a]
38      jl next
39      PRINTF fmt, [a]
40  next:
41      PRINTF fmt, [b]
42      EXIT 0
```

Select one:

- Outputs nothing
- Outputs 10
- Outputs 20
- Outputs 10 and 20

Question **4**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
30  section .data
31  fmt     db "%c",10,0
32  c       db 'A'
33  n       dq 5
34
35  section .text
36      global _start
37      extern printf
38
39  _start:
40      mov rcx, [n]
41  next:
42      push rcx
43      PRINTF fmt, [c]
44      pop rcx
45      loop next
46      EXIT 0
```

Select one:

- Outputs nothing
- Outputs A 4 times
- Outputs A 5 times
- Outputs A in an infinite loop

Question **5**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
28  section .data
29  fmt     db "%lld",10,0
30
31  section .text
32      global _start
33      extern printf
34
35  _start:
36      mov rcx, 7
37  repeat:
38      push rcx
39      PRINTF fmt, rcx
40      pop rcx
41      cmp rcx, 5
42      loopne repeat
43      EXIT 0
```

Select one:

- Outputs nothing
- Outputs 7, 6, 5, 4, 3, 2 and 1
- Outputs 7, 6, 5 and 4
- Outputs 7, 6, and 5
- Outputs 7 and 6
- Outputs 7

Started on	Monday, November 29, 2021, 11:10 AM
State	Finished
Completed on	Monday, November 29, 2021, 11:20 AM
Time taken	9 mins 56 secs
Points	3.00/3.00
Grade	**100.00** out of 100.00

Question **1**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
27  section .data
28  fmt     db "%lld",10,0
29  a       dq 0,1,2,3,4,5,6,7,8,9
30  n       dq 5
31
32  section .text
33      global _start
34      extern printf
35
36  _start:
37      mov     rcx, [n]
38  next:
39      mov     rax, [a+8]
40      push    rcx
41      PRINTF fmt, rax
42      pop     rcx
43      loop    next
44
45      EXIT    0
```

Select one:

- Outputs nothing
- Outputs 1 five times: 1, 1, 1, 1, 1
- Outputs 1 four times: 1, 1, 1, 1
- Outputs 8 five times: 8, 8, 8, 8, 8
- Outputs 8 four times: 8, 8, 8, 8
- None from the above

Question **2**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
section .data
fmt     db "%lld",10,0
a       dq 0,1,2,3,4,5,6,7,8,9
        dq 10,11,12,13,14,11,15,16,17,18,19

section .text
    global _start
    extern printf

_start:

    mov     rax, [a+8]
    PRINTF fmt, rax

    mov     rax, [a+16]
    PRINTF fmt, rax

    EXIT    0
```

Select one:

- Outputs nothing
- Outputs 1 and 2
- Outputs 8 and 16
- None from the above

Question **3**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
section .data
fmt     db "%lld",10,0
a       dq 0,1,2,3,4,5,6,7,8,9
n       dq 5

section .text
    global _start
    extern printf

_start:
    mov     rcx, [n]
    mov     rbx, a
next:
    mov     rax, [rbx+rcx*8]
    push    rcx
    PRINTF fmt, rax
    pop     rcx
    loop    next

    EXIT    0
```

Select one:

- Outputs nothing
- Outputs 0, 1, 2, 3 and 4
- Outputs 4, 3, 2, 1 and 0
- Outputs 1, 2, 3 and 4
- Outputs 4, 3, 2 and 1
- None from the above

Started on	Wednesday, December 1, 2021, 11:06 AM
State	Finished
Completed on	Wednesday, December 1, 2021, 11:10 AM
Time taken	4 mins 14 secs
Points	3.00/3.00
Grade	**100.00** out of 100.00

Question **1**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
section .data
fmt     db "%c",10,0
a       db '0','1','2','3','4','5','6','7','8','9'
n       db 5

section .text
    global _start
    extern printf

_start:
    mov     rcx, [n]
    mov     rax, a
next:
    push    rcx
    PRINTF fmt, [rax]
    pop     rcx
    inc     rax
    loop    next

    EXIT    0
```

Select one:

- Outputs nothing
- Outputs 5, 4, 3, 2 and 1
- Outputs 1, 2, 3, 4 and 5
- Outputs 4, 3, 2, 1, and 0
- Outputs 0, 1, 2, 3, and 4
- None from the above

Question **2**
Correct
1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
section .data
fmt     db "%lld",10,0
a       dq 0,1,2,3,4,5,6,7,8,9
n       dq 5

section .text
    global _start
    extern printf

_start:
    mov   rcx, [n]
    xor   rbx, rbx

next:
    add   rbx, 8
    mov   rax, [a+rbx]
    push  rcx
    PRINTF fmt, rax
    pop   rcx
    loop  next

    EXIT  0
```

Select one:

- Outputs nothing
- Outputs 0, 1, 2, 3 and 4
- Outputs 4, 3, 2, 1 and 0
- Outputs 1, 2, 3, 4 and 5
- Outputs 5, 4, 3, 2 and 1
- None from the above

Question **3**

Correct

1.00 points out of 1.00

Assume that PRINTF and EXIT are defined somewhere to printf and exit from the program. What is the output of the following code?

```
27  section .data
28  fmt     db "%lld",10,0
29  a       dq 0,1,2,3,4,5,6,7,8,9
30          dq 10,11,12,13,14,11,15,16,17,18,19
31
32  section .text
33      global _start
34      extern printf
35
36  _start:
37      mov     rbx, a
38      add     rbx, 8
39      mov     rax, [rbx]
40      PRINTF fmt, rax
41
42      add     rbx, 8
43      mov     rax, [rbx]
44      PRINTF fmt, rax
45
46      EXIT    0
```

Select one:

- Outputs nothing
- Outputs 1 and 2
- Outputs 8 and 16
- Outputs 2 and 1
- Outputs 16 and 8
- None from the above

Q1

```
section .data
format db 'rsi=%ld data=%ld',10,0
data   dq 5678

section .text
    global _start
    extern printf

_start:
    mov rdi, format     ; 1st argument (by convention)
    mov rsi, 1234       ; 2nd argument
    mov rdx, [data]     ; 3rd argument (value by address)
    xor rax, rax        ; required (no xmm registers used)
    call printf

    mov rax, 60     ; syscall number for exit
    mov rdi, 0      ; int status
    syscall
```

Output: _____

Q2

```
section .data
format db 'x+y=%ld',10,0
x   dq 10
y   dq 42

section .text
    global _start
    extern printf

_start:
    mov rdi, format
    mov rsi, [x]
    add rsi, [y]
    xor rax, rax
    call printf

    mov rax, 60     ; syscall number for exit
    mov rdi, 0      ; int status
    syscall
```

Output: 52_____

Q3

```
section .data
format db 'x*y=%ld',10,0
x   dq 10
y   dq 42

section .text
    global _start
    extern printf

_start:
    mov rdi, format
    mov rsi, [x]
    imul rsi, [y]
    xor rax, rax
    call printf

    mov rax, 60     ; syscall number for exit
    mov rdi, 0      ; int status
    syscall
```

Output: 420_____

Q4

```
10  section .data
11
12  format db '%lld',10,0
13
14  section .text
15      global _start
16      extern printf
17
```

```
17
18  _start:
19      jmp     next        ; unconditional jump
20
21      ; output
22      mov     rdi, format
23      mov     rsi, 10
24      xor     rax, rax
25      call    printf
26
27  next:
28      mov rax, 60         ; syscall number for exit
29      mov rdi, 0          ; int status 0
30      syscall
```

Output: _____

Q5

```
8   section .data
9
10  true  db "true",10,0
11  false db "false",10,0
12
13  section .text
14      global _start
15      extern puts
```

```
17  _start:
18      mov     rcx, 5
19      mov     rdx, 5
20      cmp     rcx, rdx
21      je      equal       ; conditional jump
22
23      ; Output false
24      mov     rdi, false
25      xor     rax, rax
26      call    puts
27      jmp     end
28
29  equal:
30      ; Output true
31      mov     rdi, true
32      xor     rax, rax
33      call    puts
34
35  end:
36      mov rax, 60         ; syscall number for exit
37      mov rdi, 0          ; int status 0
38      syscall
39
```

Output: _____

Q6

```
8   section .data
9   a    dq 10
10  b    dq 20
11  str1 db "10<20",10,0
12  str2 db "10>20",10,0
13
14  section .text
15      global _start
16      extern printf
```

```
18  _start:
19      mov  rax, [a]
20      cmp  rax, [b]
21      jg   next
22
23      mov  rdi, str1
24      xor  rax, rax
25      call printf
26      jmp  end
27
28  next:
29      mov  rdi, str2
30      xor  rax, rax
31      call printf
32
33  end:
34      mov rax, 60     ; syscall number for exit
35      mov rdi, 0      ; int status 0
36      syscall
```

Output: _____

Q7

```
6   section .data
7   str1 db "loop",0
8
9   section .text
10      global _start
11      extern puts
```

```
13  _start:
14      mov  rcx, 3
15
16  next:
17      push rcx        ; Save rcx (puts uses it too)
18
19      mov  rdi, str1
20      call puts
21
22      pop  rcx        ; Restore rcx
23
24      dec  rcx
25      jnz  next
26
27      mov rax, 60     ; syscall number for exit
28      mov rdi, 0      ; int status 0
29      syscall
```

Output: _____

Q8

```
6   section .data
7   str1 db "%lli ",10,0
8
9   section .text
10      global _start
11      extern printf
12
```

```
13  _start:
14      mov rcx, 5      ; Set counter to 5
15
16  repeat:
17      push rcx        ; Save
18
19      ; Output
20      mov rdi, str1
21      mov rsi, rcx
22      xor rax, rax
23      call printf
24
25      pop rcx         ; Restore
26      loop repeat
27
28      mov rax, 60     ; syscall number for exit
29      mov rdi, 0      ; int status 0
30      syscall
```

Output: _____

Q9

```
6   section .data
7   fmt db "%lli",10,0
8
9   section .text
10      global _start
11      extern printf
```

```
13  _start:
14      mov rcx, 10     ; Set counter to 10
15
16  repeat:
17      push rcx        ; Save
18
19      ; Output
20      mov rdi, fmt
21      mov rsi, rcx
22      xor rax, rax
23      call printf
24
25      pop rcx         ; Restore
26
27      cmp rcx, 5
28      loopne repeat
29
30      mov rax, 60     ; syscall number for exit
31      mov rdi, 0      ; int status 0
32      syscall
```

Output: _____

FYI:

```
9   section .data
10      db      0x55                ; just the byte 0x55
11      db      0x55,0x56,0x57      ; three bytes in succession
12      db      'a',0x55            ; character constants are OK
13      db      'hello',13,10,'$'   ; so are string constants
14      dw      0x1234              ; 0x34 0x12
15      dw      'a'                 ; 0x41 0x00 (it's just a number)
16      dw      'ab'                ; 0x41 0x42 (character constant)
17      dw      'abc'               ; 0x41 0x42 0x43 0x00 (string)
18      dd      0x12345678          ; 0x78 0x56 0x34 0x12
19      dq      0x1122334455667788  ; 0x88 0x77 0x66 0x55 0x44 0x33 0x22 0x11
20      dd      1.234567e20         ; floating-point constant
21      dq      1.234567e20         ; double-precision float
22      dt      1.234567e20         ; extended-precision float
```

Q10

```
1  ; Str.
2  ; Set elements of array (as a null-terminated string)
3  ; Run: $ nasm -f elf64 str.asm && ld -dynamic-linker /li
4  ; Output: ABC
5
6  %include "macros.inc"
7
8  section .data
9  arr times 10 db 0   ; reserve 10 bytes and fill with 0
10 fmt db "%s",10,0
11
12 section .text
13     global _start
14     extern printf
15
16 _start:
17     mov     rax, arr
18     mov byte [rax], 'A'
19     mov byte [rax+1], 'B'
20     mov byte [rax+2], 'C'
21
22     PRINTF fmt, arr
23     EXIT
```

Output: _____

Q11

```
8  %include "macros.inc"
9
10 section .data
11 src   db '01234567890123456789012345678 9',10,0
12 dst   times 30 db 0
13 fmt   db "%s",10,0
```

```
15 section .text
16     global _start
17     extern printf
18
19 _start:
20     mov rcx, 3
21 repeat:
22     ; src -/-> dst, so src --> reg --> dst
23     mov rax, [src+(rcx-1)*8]
24     mov [dst+(rcx-1)*8], rax
25     loop repeat
26
27     PRINTF fmt, dst
28     EXIT
```

Output: _____

Q12

```
%macro PRINTF 2
    push rdi
    push rsi
    push rax
    mov rdi, %1
    mov rsi, %2
    xor rax, rax
    call printf
    pop rax
    pop rsi
    pop rdi
%endmacro
```

```
section .data
fmt     db "%lld",10,0
n       dq 30

section .text
    global _start
    extern printf

_start:
    mov rax, 10
    PRINTF fmt, rax
    PRINTF fmt, 20
    PRINTF fmt, [n]
    EXIT
```

Output: _____

Q13

Cont.

```
%include "macros.inc"

section .data
fmt     db "%lld",10,0
n       dq 30

section .text
    global _start
    extern printf
    extern time

_start:
    mov     rdi, 0
    call    time
    PRINTF fmt, rax
    EXIT
```

```
%macro PRINTF 2
    push rdi
    push rsi
    push rax
    mov rdi, %1
    mov rsi, %2
    xor rax, rax
    call printf
    pop rax
    pop rsi
    pop rdi
%endmacro

%macro EXIT 0
    mov rax, 60
    mov rdi, 0
    syscall
%endmacro
```

Output: _____

Q14

```
; Time
; Output time in seconds since 1 Jan 1970
; Run: $ nasm -f elf64 time.asm && ld -dyna

%include "macros.inc"

section .data
fmt     db "%lld",10,0

section .text
    global _start
    extern printf
    extern time

_start:
    mov     rdi, 0
    call    time
    PRINTF fmt, rax
    EXIT
```

Output: _____

Q15

rand.asm

```
; Random number generator using c's rand().
; rand()'s result is the number
; Run: $ nasm -f elf64 rand.asm && ld -dynamic-linker /lib(

%include "macros.inc"

section .data
fmt     db "%lld",10,0

section .text
    global _start
    extern printf
    extern time
    extern srand
    extern rand

_start:
    ; Get the current time
    xor   rdi, rdi
    call  time

    ; Seed the random number generator
    mov   rdi, rax
    call  srand

    ; Get the random number
    call  rand

    ; Map the number into range [0,100)
    mov   rdx, 0
    mov   rbx, 100
    idiv  rbx   ; Note, remainder in rdx

    PRINTF fmt, rdx
    EXIT
```

Output: _____

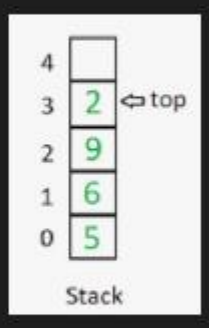
Q16

```
section .data
fmt     db "%lld",10,0

section .text
    global _start
    extern printf

_start:
    mov   rax, 6
    mov   rbx, 9
    mov   rcx, 2
    push  rax
    push  rbx
    push  rcx
    pop   rax
    pop   rbx
    pop   rcx

    PRINTF fmt, rax
    EXIT
```



Output: _____

Q17

```
11  section .text
12      global _start
13      extern printf
14
15  _start:
16      mov     rax, 1
17      call    foo
18      PRINTF fmt, rax
19      jmp     end
20
21  foo:
22      mov     rax, 2
23      ret
24
25  end:
26      EXIT
```

Output: _____

Q18

```
11  section .text
12      global _start
13      extern printf
14
15  _start:
16      mov     rax, 1
17      push    rax
18      call    foo
19      pop     rax
20      PRINTF fmt, rax
21      jmp     end
22
23  foo:
24      mov     rax, 2
25      ret
26
27  end:
28      EXIT
```

Output: _____

Q19

```
15  _start:
16      push    rax
17      push    rbx
18      push    rcx
19
20      mov     rax, 1
21      mov     rbx, 2
22      call    add
23
24      PRINTF fmt, rcx
25
26      pop     rcx
27      pop     rbx
28      pop     rax
29
30      jmp     end
```

```
32  add:
33      add     rbx, rax
34      mov     rcx, rbx
35      ret
36
37  end:
38      EXIT
```

Output: _____

Q20

```
15  _start:
16      push    rax
17      push    rbx
18      push    rcx
19
20      push    1
21      push    2
22      call    add
23      add     rsp, 16  ; instead of 2 pops
24
25      PRINTF fmt, rcx
26
27      pop     rcx
28      pop     rbx
29      pop     rax
30
31      jmp     end
32
33  add:
34      mov     rax, [rsp+8]
35      add     rax, [rsp+16]
36      mov     rcx, rax
37      ret
38
39  end:
40      EXIT
```

Output: _____

Q21

```
15  _start:
16      push    rax
17      push    rbx
18      push    rcx
19
20      push    1
21      push    2
22      call    add
23      add     rsp, 16  ; instead of
24
25      PRINTF fmt, rcx
26
27      pop     rcx
28      pop     rbx
29      pop     rax
30
31      jmp     end
33  add:
34      ; Prologue
35      push rbp
36      mov  rbp, rsp
37
38      mov     rax, [rbp+16]
39      add     rax, [rbp+24]
40      mov     rcx, rax
41
42      ; Epilogue
43      mov rsp, rbp
44      pop rbp
45
46      ret
47
48  end:
49      EXIT
```

Output: _____