Started on	Thursday, 24 June 2021, 2:00 PM
State	Finished
Completed on	Thursday, 24 June 2021, 2:44 PM
Time taken	43 mins 9 secs
Grade	**0.00** out of 36.00 (**0**%)

Question 1

Incorrect

Mark 0.00 out of 2.00

Assume that the registers rax, rbx, and rcx initially have value

%rax = 0x011a

%rbx = 0x011b

%rcx = 0x011c

Assume the memory locations 0x011a, 0x011b, and 0x011c store the values 0x022a, 0x022b, 0x022c respectively.

We run the following operations on these registers:

movq %rax, %rbx

movq %rbx, (%rcx)

leaq (%rcx), %rax

movq (%rax), %rax

What is the value in %rax after this?

a. 0x022c

b. 0x011c

c. 0x011b

d. 0x022a ✖

e. 0x022b

f. 0x011a

Your answer is incorrect.

The correct answer is:
0x011a

Question 2

Not answered

Marked out of 2.00

```c
#include<stdio.h>
int no_of_factors(int n);
int main(){
    int n,count;
    scanf("%d",&n);
    count=no_of_factors(n);
    printf("%d\n",count);
}

```

```
.text
.global no_of_factors
no_of_factors:
    movq $0 ,%rcx
    movq $0 ,%rbx
.loop:
    cmpq %rdi,%rbx
    je .break
    inc %rbx
    movq $0 ,%rdx
    movq %rdi ,%rax
    idiv %rbx
    cmpq $1 ,%rdx
    je .increase
    jmp .loop
.increase:
  inc %rcx
  jmp .loop
.break:
  movq %rcx,%rax
  ret

```

This code is intended to give the number of factors of a given input, but there is an error in the code. Which of the possible changes will ensure that code works as required?

- a. Changing line 14 to jle .increase
- b. Changing line 13 to cmpq $0, %rax
- c. Changing line 13 to cmpq $0, %rdx
- d. Changing line 14 to jl .increase

Your answer is incorrect.

The correct answers are:
Changing line 13 to cmpq $0, %rdx,

Changing line 14 to jl .increase

Question 3

Not answered

Marked out of 3.00

Consider the below assembly code:

```
.text
.global main

main:
    mov     N@GOTPCREL(%rip),   %rdx
    mov     (%rdx), %rdx
    mov     $0, %rbx
    mov     $1, %rax
    mov     $1, %rcx

L0:
    cmp     %rdx,   %rcx
    je      L1
    mov     %rax,   %r8
    add     %rbx,   %rax
    mov     %r8,    %rbx
    inc     %rcx
    jmp     L0

L1:
    mov     $10,    %rbx
    mov     $0,     %rcx

L2:
    cmp     $0,     %rax
    je      .exit
    cqto
    idivq   %rbx
    add     %rdx,   %rcx
    jmp     L2

.exit:
    mov     %rcx,   %rax
    ret
```

If the value of N in the data section is 13, what will be returned from the above assembly code?

- a. 8
- b. 26
- c. 65
- d. 4

Your answer is incorrect.

The correct answer is:
8

Question 4

Not answered

Marked out of 2.00

Let %ebp = 0xA1C. Assume function fun is called from inside another function. At what memory location within the stack frame of the caller function, will the variable X be located. Integers are 4 byte long. Assume all the function arguments are passed using the stack.

```
void fun(int A, int B, int C, int X, int Y)
{
        // do something
        return;
}
```

a. 0xA34

b. 0xA30

c. 0xA32

d. 0xA31

Your answer is incorrect.

The correct answer is:
0xA30

Question 5

Not answered

Marked out of 2.00

Find the value stored in register %r12

```
movq $10, %r8
movq $20, %r9
movq $30, %r10
movq $40, %r11
movq $50, %r12
andq %r8, %r9
addq %r9, %r10
xorq %r10, %r11
subq %r11, %r12
addq %r8, %r12
subq %r12, %r12
subq %r12, %r9
addq %r12, %r10
notq %r11
addq %r12, %r11
```

Answer:

The correct answer is: 0

Question 6

Not answered

Marked out of 2.00

Find the value stored in registers %rdx

```
movq $9, %r8
movq $6, %r9
movq $2, %r10
movq %r9, %rdx
subq %r10, %rdx
movq %rdx, %rax
salq $63, %rax
sarq $63, %rax
imulq %r8, %rdx
xorq %rdx, %rax
```

Answer: ✖

The correct answer is: 36

Question 7

Incorrect

Mark 0.00 out of 2.00

Assume function 1 is located at address 0xC1D3 and is called from inside function 2. If the present instruction is a call instruction to function 1. What will be the value of %esi, %esp, and M[%esp+4], after the call instruction executes. Here M[x] refers to the value stores at address x in memory. Given %esi = 0xB1D3, %esp = 0xC2D2 and call instructions are 4 bytes long.

a. %esi = 0xB1D7, %esp = 0xC1D3, %M[%esp+4] = 0xC2CE

b. %esi = 0xC1D3, %esp = 0xC2CE, %M[%esp+4] = 0xB1D7

c. %esi = 0xC2CE %esp = 0xC1D3, %M[%esp+4] = 0xB1D7 ✖

d. %esi = 0xC1D3, %esp = 0xB1D7, %M[%esp+4] = 0xC2CE

Your answer is incorrect.

The correct answer is:
%esi = 0xC1D3, %esp = 0xC2CE, %M[%esp+4] = 0xB1D7

Question 8

Not answered

Marked out of 2.00

Given below are 3 code snippets of x86-64 assembly functions

Function-1:

```
f1:
   xorl    %eax, %eax
L2:
   movsbq  (%rdi), %rdx
   subq    $48, %rdx
   cmpq    $9, %rdx
   ja      L5
   imulq   $10, %rax, %rax
   incq    %rdi
   addq    %rdx, %rax
   jmp     L2
L5:
   ret
```

Function-2:

```
f2:
   movq    %rdi, %rax
L7:
   cmpb    $0, (%rax)
   je      L9
   incq    %rax
   jmp     L7
L9:
   cmpq    %rax, %rdi
   jnb     L11
   decq    %rax
   movb    (%rdi), %cl
   incq    %rdi
   movb    (%rax), %dl
   movb    %cl, (%rax)
   movb    %dl, -1(%rdi)
   jmp     L9
L11:
   ret
```

Function-3:

```
f3:
   xorl    %eax, %eax
L13:
   cmpq    %rax, %rdx
   je      L15
   movb    (%rdi,%rax), %cl
   movb    (%rsi,%rax), %r8b
   movb    %r8b, (%rdi,%rax)
   movb    %cl, (%rsi,%rax)
   incq    %rax
   jmp     L13
L15:
   ret
```

Let A denote the number of functions described above that modify memory.

Let B denote the number of functions described above that never modify any caller-saved register.

Let X2 denote the number of arguments that Function-2 takes, assuming that there are no unused arguments.

Let X3 denote the number of arguments that Function-3 takes, assuming that there are no unused arguments.

Calculate the value of A + B + X2 + X3

Answer:

The correct answer is: 6

Question 9

Not answered

Marked out of 2.00

Consider the algorithm described below.

```
fun:
    movq %rdi, %r8
    movq %rsi, %r9
    movq $1, %rbx

.L1:
    movq $1, %r12
    andq %r9, %r12
    cmpq $0, %r12
    jz .L2
    imulq %r8, %rbx

.L2:
    imulq %r8, %r8
    sarq $1, %r9
    cmpq $0, %r9
    jne .L1

.L6:
    movq %rbx, %rax
    ret
```

Assume that a is present in %rdi and n is present in %rsi. Also, a is a positive integer and n is a non-negative integer. The function which represents the above algorithm is:

a. $\begin{cases} 0, & n = 0 \\ a.\left(a^{\frac{n}{2}}\right)^2, & n > 0 \text{ and } n \text{ is even} \\ \left(a^{\frac{n-1}{2}}\right)^2, & n > 0 \text{ and } n \text{ is odd} \end{cases}$

b. $\begin{cases} 1, & n = 0 \\ \left(a^{\frac{n}{2}}\right)^2, & n > 0 \text{ and } n \text{ is even} \\ a.\left(a^{\frac{n-1}{2}}\right)^2, & n > 0 \text{ and } n \text{ is odd} \end{cases}$

c. $\begin{cases} 0, & n = 0 \\ \left(a^{\frac{n}{2}}\right)^2, & n > 0 \text{ and } n \text{ is even} \\ a.\left(a^{\frac{n-1}{2}}\right)^2, & n > 0 \text{ and } n \text{ is odd} \end{cases}$

d. None of the above

e. $\begin{cases} 1, & n = 0 \\ a.\left(a^{\frac{n}{2}}\right)^2, & n > 0 \text{ and } n \text{ is even} \\ \left(a^{\frac{n-1}{2}}\right)^2, & n > 0 \text{ and } n \text{ is odd} \end{cases}$

Your answer is incorrect.

The correct answer is:

$\begin{cases} 1, & n = 0 \\ \left(a^{\frac{n}{2}}\right)^2, & n > 0 \text{ and } n \text{ is even} \\ a.\left(a^{\frac{n-1}{2}}\right)^2, & n > 0 \text{ and } n \text{ is odd} \end{cases}$

Question 10

Not answered

Marked out of 1.00

Find the value stored in registers %rdx and %rax.

Ensure that the values are in the format %rdx, %rax. For example, if %rdx = 1, %rax = 2, write it as 1, 2

```
movq $16, %r8
movq $2, %r9
movq %r8, %rdx
movq %rdx, %rax
sarq $63, %rdx
idivq %r9
imulq %r8
```

Answer:

The correct answer is: 0, 128

Question 11

Not answered

Marked out of 3.00

Consider the below assembly code:

```
main:
    mov     N@GOTPCREL(%rip),   %rax
    mov     (%rax),             %rax
    mov     $0,                 %rbx
    mov     $0,                 %rcx
    mov     $0,                 %r10
    mov     $10,                %r8

L0:
    cmp     $0,                 %rax
    je      L3
    cqto
    idivq   %r8
    cmp     $0,                 %r10
    je      L1
    jne     L2
    add     %rdx,               %rbx

L1:
    add     %rdx,               %rcx
    inc     %r10
    jmp     L0

L2:
    add     %rdx,               %rbx
    dec     %r10
    jmp     L0

L3:
    sub     %rbx,               %rcx
    mov     %rcx,               %rax
    cmp     $0,                 %rax
    jge     .exit

L4:
    neg     %rax

.exit:
    ret
```

If the value of N in the data section is 1458291, what will be returned from the above assembly code?

- a. 11
- b. None
- c. 28
- d. 5

Your answer is incorrect.

The correct answer is:
None

Question 12

Not answered

Marked out of 1.00

Assume that the base of the stack used in an assembly program is at 0x0067 (this is truncated, assume it to be a valid memory location). What will be the starting address of the 11th element on the stack (stack is 0 indexed)?

a. 0x0007

b. 0x000F

c. None of the above

d. 0x00BF

Your answer is incorrect.

The correct answer is:
0x0007

Question 13

Not answered

Marked out of 2.00

Consider the following assembly function described below.

```
fun:
      pushl %ebp
      movl %esp, %ebp
      movl $0, %ecx
      cmpl $11, %edx
      jne .L2
      movl $4, %ecx
      jmp .L3
.L2:
      cmpl $22, %edx
      jne .L3
      movl $7, %ecx
.L3:
      cmpl $55, %edx
      jne .L5
      movl $7, %ecx
.L5:
      cmpl $33, %edx
      sete %al
      cmpl $44, %edx
      sete %dl
      orl %edx, %eax
      testb $1, %al
      je .L6
      movl $11, %ecx
.L6:
      movl %ecx, %eax
      popl %ebp
      ret
```

Assume that the register value in %edx is 22, then what value will be returned from this function.

Answer:

The correct answer is: 7

Question 14

Not answered

Marked out of 2.00

Which registers are always used in the execution of a C program, assuming 64 bit architecture?

- a. %rax
- b. %rip
- c. %rsp
- d. None
- e. %rbp

Your answer is incorrect.

The correct answers are:
%rsp,

%rbp,

%rip

Question 15

Not answered

Marked out of 2.00

```c
#include<stdio.h>
int sumN(int N);
int main(){
   int n;
   scanf("%d",&n);
   printf("%d\n",sumN(n));
}

```

```
.text
.global sumN
sumN:
   pushq %rbx
   movq %rdi,%rbx
   movq $0 ,%rax
   cmpq $0 ,%rdi
   jl .base
   leaq -1(%rdi),%rdi
   call sumN
   addq %rbx,%rax
.base:
   popq %rbx
   ret
```

This code is intended to find the sum of n natural numbers where n is passed as input, but there is an error in the code, which of the possible changes will ensure that code works as required.

a. Change line 8 to jle .base

b. Change line 8 to jl .base

c. Change line 8 to je .base

d. Change line 8 to jg .base

Your answer is incorrect.

The correct answers are:
Change line 8 to jle .base,

Change line 8 to je .base

Question 16

Not answered

Marked out of 2.00

```
1   .data
2   num:
3   .quad 8
4   .comm array,64,64
5   sum:
6   .quad
7   max:
8   .quad
9   .text
10  .global main
11  main:
12      movq array@GOTPCREL(%rip),%rax
13      movq $2,array(%rip)
14      movq $8,array+8(%rip)
15      movq $3,array+16(%rip)
16      movq $7,array+24(%rip)
17      movq $5,array+32(%rip)
18      movq $14,array+40(%rip)
19      movq $1,array+48(%rip)
20      movq $6,array+56(%rip)
21      movq $0,%rcx
22      movq $0,%rsi
23      movq (%rax),%rdi
24      movq num@GOTPCREL(%rip),%r8
25      movq (%r8),%r9
26  .loop:
27      leaq (%rax,%rcx,8),%rdx
28      movq (%rdx),%rbx
29      cmpq %rdi,%rbx
30      cmovge %rbx,%rdi
31      dec %rcx
32      cmpq %r9,%rcx
33      jne .loop
34      je .exit
35  .exit:
36      movq max@GOTPCREL(%rip),%r8
37      movq %rdi,(%r8)
38      ret
39
```

Code given is to find the maximum value in a given array, but this does not work well, try to figure out the error(s) in the code.

You are expected to write the line numbers corresponding to the code where you feel that there is an error. (For example, if there are errors in line 5 and 6, then the answer is L5, L6 ; if the error is only in line 8, then the answer is L8)

Answer:

The correct answer is: L31

Question 17

Not answered

Marked out of 1.00

Find the value stored in register %rax:

```
movq $5, %r8
movq $8, %r9
movq $9, %r10
movq %r9, %rax
xorq %r8, %rax
sarq $3, %rax
notq %rax
subq %r10, %rax
```

Answer:

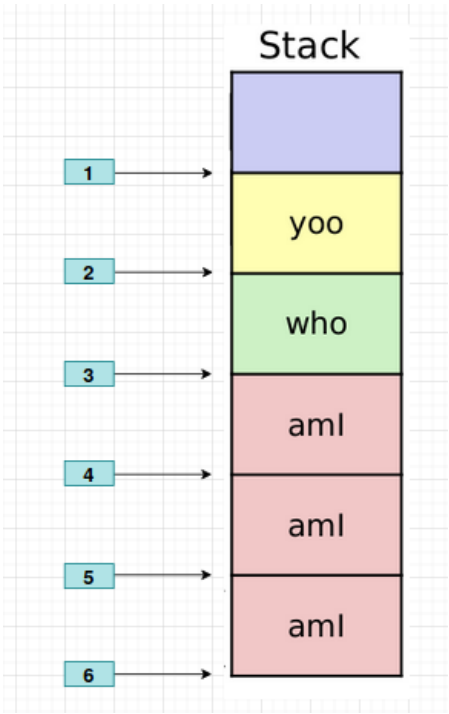
The correct answer is: -11

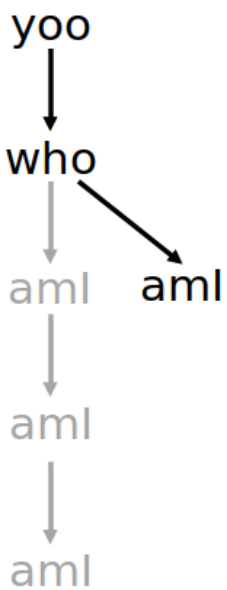
Question 18

Not answered

Marked out of 1.00

yoo is the function that calls who and who calls aml which is a recursive function. Let us assume we are at highlighted aml function, where will %rbp and %rsp point to in the stack image provided respectively.





a. 3 and 4

b. 4 and 3

c. 3 and 2

d. 2 and 3

Your answer is incorrect.

The correct answer is:
3 and 4

Question 19

Not answered

Marked out of 1.00

```c
#include<stdio.h>
void strev(char *arr,int size);
int main(){
    int n;
    printf("enter the size of the string\n");
    scanf("%d",&n);
    char c[n];
    printf("enter the string\n");
    scanf("%s",c);
    printf("entered string is %s\n",c);
    strev(c,n);
    printf("reversed string is %s\n",c);
}
```

```
.text
.global strev
strev:
movl %esi,%r9d
dec %r9d
movl $0 ,%r8d
.loop:
cmp %r8d,%r9d
jl .exit
leaq (%rdi,%r8,4),%r10
movb (%r10),%r12b
leaq (%rdi,%r9,1),%r13
movb (%r13),%r11b
movb %r11b,(%r10)
movb %r12b,(%r13)
inc %r8d
dec %r9d
jmp .loop
.exit:
ret
```

This code is intended to reverse a given string, but there is an error in the code, which of the possible changes will ensure that code works as required.

a. Change line 12 to leaq (%rdi,%r9,4),%r10

b. Change line 10 to leaq (%rdi,%r8,1),%r10

c. Change line 12 to leaq (%rdi,%r9,2),%r10

d. Change line 10 to leaq (%rdi,%r8,2),%r10

Your answer is incorrect.

The correct answer is:
Change line 10 to leaq (%rdi,%r8,1),%r10

Question 20

Not answered

Marked out of 1.00

Assume the following function written in C:

void hello(int a, int c, int b, int d)

Based on this definition, where will variable b be stored?

Answer:

The correct answer is: %rdx