Question 1

17.14 / 20 pts

Consider the following machine code:

```
(gdb) disass main
Dump of assembler code for function main:
  0x080483db <+0>:    push   %ebp
  0x080483dc <+1>:    mov    %esp,%ebp
  0x080483de <+3>:    sub    $0x10,%esp
  0x080483e1 <+6>:    movl   $0x0,-0x10(%ebp)
  0x080483e8 <+13>:    movl   $0x0,-0xc(%ebp)
  0x080483ef <+20>:    movl   $0xc,-0x8(%ebp)
  0x080483f6 <+27>:    movl   $0x64,-0x4(%ebp)
  0x080483fd <+34>:    mov    -0x10(%ebp),%edx
  0x08048400 <+37>:    mov    -0x8(%ebp),%eax
  0x08048403 <+40>:    add    %edx,%eax
  0x08048405 <+42>:    mov    %eax,-0xc(%ebp)
  0x08048408 <+45>:    shll   $0x3,-0x8(%ebp)
  0x0804840c <+49>:    movl   $0xa0000,-0x10(%ebp)
  0x08048413 <+56>:    mov    -0xc(%ebp),%eax
  0x08048416 <+59>:    sub    -0x10(%ebp),%eax
  0x08048419 <+62>:    mov    %eax,-0x4(%ebp)
  0x0804841c <+65>:    mov    -0x4(%ebp),%eax
  0x0804841f <+68>:    add    %eax,-0x8(%ebp)
  0x08048422 <+71>:    mov    -0x8(%ebp),%eax
  0x08048425 <+74>:    leave
  0x08048426 <+75>:    ret
```

Making reasonable assumptions from the above machine code, fill in the blanks with appropriate values in the following C program:

(To represent numeric values, only use decimal format (.ie. base10) without any spaces)

```
int main(){
   int w = 0, x = 0, y = 12, z =
[ Select ]
 ;
   x =
[ Select ]
 + y;
   y = y <<
[ Select ]
 ;
```

```
   w = 10 * (2 <<
[ Select ]
 );
   z =
[ Select ]
 - w;

[ Select ]
 = z + y;
   return
[ Select ]
 ;
}
```

Answer 1:
Correct!
100
Answer 2:
Correct!
w
Answer 3:
Correct!
3
Answer 4:
Correct Answer
15
You Answered
17
Answer 5:
Correct!
x
Answer 6:
Correct!
y
Answer 7:
Correct!
y

Question 2
13.33 / 20 pts
Consider the binary https://bit.ly/3HOMEWORK1 (Links to an external site.) and answer the following questions:

In case you cannot access the shortened link for the binary, use the following google drive link (They both redirect you to the same binary)
https://drive.google.com/file/d/1VdRDA37Inq0Rw6U__47v-VBzhhoTKjAz/view (Links to an external site.)

1. How many times does the statement main+34 execute?

[Select]

2. What do the operations main+45 and main+48 evaluate to?

[Select]

3. What value should have been used in line main+34 to make the program return 75?

[Select]

Answer 1:
Correct!
2
Answer 2:
Correct!
eax=eax*15
Answer 3:
You Answered
7
Correct Answer
4

Question 3
10 / 10 pts
A legacy system hosting a static website was compromised by an attacker. The forensics team identified that the attacker was running the following binary on the system.

https://bit.ly/homework1question3 (Links to an external site.)

In case you cannot access the shortened link for the binary, use the following google drive link (They both redirect you to the same binary)

https://drive.google.com/file/d/1JprGgGL3lrrrAYJ4HOGUw9aa6JYxOuN4/view (Links to an external site.)

Which of the following options do you think best describes the attacker's motive?

(This binary will not damage your System/VM, I promise.)

The attacker might be trying to make a remote connection back to his server to gain a remote shell to the system.
Correct!
The attacker might be trying to exhaust CPU resources by performing multiple CPU operations that may lead to Denial of Service attack.
The attacker might be trying to exhaust memory of the system by sequentially writing in successive memory blocks that may lead to Denial of Service attack.
The attacker might be attempting to perform multiple I/O operations on disk to exhaust the disk usage that may lead to Denial of Service attack.

Question 4
10 / 15 pts
Match each of the assembler routines with the equivalent C function.

```
foo1:
   pushl %ebp
   movl %esp,%ebp
   movl 8(%ebp),%eax
   sall $4,%eax
   subl 8(%ebp),%eax
   movl %ebp,%esp
   popl %ebp
   ret
foo2:
   pushl %ebp
   movl %esp,%ebp
   movl 8(%ebp),%eax
   testl %eax,%eax
   jge .L4
   addl $15,%eax
.L4:
   sarl $4,%eax
   movl %ebp,%esp
   popl %ebp
   ret


foo3:
   pushl %ebp
   movl %esp,%ebp
   movl 8(%ebp),%eax
   shrl $31,%eax
   movl %ebp,%esp
   popl %ebp
   ret
int choice1(int x) {
   return (x < 0);
}
int choice2(int x) {
   return (x << 31) & 1;
}
int choice3(int x) {
   return 15 * x;
}
int choice4(int x) {
   return (x + 15) /4
}
int choice5(int x) {
   return x / 16;
}
int choice6(int x) {
   return (x >> 31);
}
```

Correct!

Foo1

Choice3

Correct!

Foo2

Choice5

You Answered

Foo3

Choice6

Correct AnswerChoice1

Other Incorrect Match Options:

Choice4

Choice6

Choice2

Question 5

5 / 5 pts

Which of the following might set a conditional flag?

mov

Correct!

test

Correct!

cmpl

Correct!

add

Question 6

10 / 10 pts

1) In Two’s complement signed representation, incrementing from its maximum value gives positive value for that type.

2) In Unsigned representation, decrementing from its minimum value gives negative value for that type.

True, True

Correct!

False, False

True, False

False, True

Question 7

20 / 20 pts

Consider the following machine code:

```
  0x080483db <+0>:    push   %ebp
  0x080483dc <+1>:    mov    %esp,%ebp
  0x080483de <+3>:    sub    $0x10,%esp
  0x080483e1 <+6>:    movl   $0x64,-0x4(%ebp)
  0x080483e8 <+13>:   movl   $0x0,-0x8(%ebp)
  0x080483ef <+20>:   mov    -0x4(%ebp),%eax
  0x080483f2 <+23>:   sub    $0x63,%eax
  0x080483f5 <+26>:   mov    %eax,-0x8(%ebp)
  0x080483f8 <+29>:   jmp    0x80483fe <main+35>
  0x080483fa <+31>:   addl   $0x2,-0x8(%ebp)
  0x080483fe <+35>:   mov    -0x4(%ebp),%eax
  0x08048401 <+38>:   add    $0xa,%eax
  0x08048404 <+41>:   cmp    -0x8(%ebp),%eax
  0x08048407 <+44>:   jge    0x80483fa <main+31>
  0x08048409 <+46>:   mov    $0x0,%eax
```

```
   0x0804840e <+51>:    leave
   0x0804840f <+52>:    ret
```

Choose the correct answers for [A], [B] and [C] based on the following C program:

```
    int x = 100,y = 0;
    for(y = [A];y <= [B]; y+=[C]){
        //no code here
    }
```

C = 10

A = x-63

Correct!

B = x+10

Correct!

A = x-99

Correct!

C = 2

B = x+2