CS 218 - Worksheet 3

← Encode last 5 of NSHE ID

Enter Name: First Last

Chayden Richardson

For each of the following questions, select only the best answer.

Question 1 What is the hex, double-word size, two's compliment representation of -11?

1/1

- [ ] 0xfffffffc
- [X] 0xfffffff5
- [ ] 0x000000fc
- [ ] 0xffffffff

```
0000 1011   invert
1111 0100      ↓
        +        1

1111 0101

0xF5
```

Question 2 What is the decimal representation of 0xFFFFFFFD (hex, double-word size, two's compliment)?

1/1

- [ ] -5
- [ ] -2
- [X] -3
- [ ] -4

```
1111 1101
0000 0010
        +        1

6000 0011

0x03  => -3
```

Question 3 What is the IEEE 32-bit floating point representation of +11.125 in hex?

1/1

- [X] 0x41320000
- [ ] 0x0xfd1250
- [ ] 0x43120000
- [ ] 0xc3120000

Question 4 Given the following code fragment, what is in the **rbx** register?

```
mov     rax, 9
mov     rbx, 2
add     rbx, rax
```

1/1

- [ ] rbx = 0x0000000000000011
- [ ] rbx = 0x0000000000000002
- [ ] rbx = 0x0000000000000000
- [X] rbx = 0x000000000000000b -

Question 5 Given the following code fragment, what is in the **eax** register?

```
mov     eax, 4
mov     ebx, 7
sub     eax, ebx
```

1/1

- [ ] eax = -0x00000003
- [X] eax = 0xfffffffd -
- [ ] eax = 0x00000004
- [ ] eax = 0xfffffff3

Question 6 Given the following code fragment, what is in the **eax** and **edx** registers?

```
mov     eax, 4
mov     ebx, 3
imul    ebx
```

1/1

- [ ] eax = 0x00000012 and edx = 0x00000000
- [ ] eax = 0x00000000 and edx = 0x00000012
- [X] eax = 0x0000000c and edx = 0x00000000 -
- [ ] eax = 0x00000001 and edx = 0x00000001

Question 7 Given the following code fragment, what is in the **eax** and **edx** registers?

```
mov     eax, 5
cdq
mov     ebx, 3
idiv    ebx
```

1/1

- [ ] eax = 0x00000001 and edx = 0x00000000
- [ ] eax = 0x00000002 and edx = 0x00000000
- [ ] eax = 0x00000002 and edx = 0x00000001
- [X] eax = 0x00000001 and edx = 0x00000002 -

Question 8   What function does each of the following instructions perform.

```
mov    rax, qword [var]
mov    rax, var
```

☐ They both access the address.

☒ The first accesses the value and the second access the address.

☐ They both access the value.

☐ The first accesses the address and the second access the value.