**Name : Từ Khai Hoài**

**Student ID: 20125029**

1.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | 1’s complement | 2’s complement | Bias 10 |
| 0 | 00 | 00 | 0A |
| -7 | 18 | 19 | 03 |
| 16 | \_ | \_ | 1A |
| -14 | 11 | 12 | \_ |
| 20 | \_ | \_ | 1E |

Short explanation: For 1’s complement, the last bit is sign bit, first 4-bit of positive number is same as normal binary, negative is NOT (positive 4-bit). For bias-10, we plus 10 to the number and convert to binary number. For 2 complement, last bit is also sign bit and equal -16, we use this to calculate negative number.

2.

a.

Exponent: 18

Value of fraction: 29/32

Final decimal value: 15.25

b.

Base-2 normal notation: - 1000.(0011)

5-bit encoding of exponent: 10010

6-bit encoding of fraction: 000010

c.

Negative infinity: 0xFA0

Largest denomalized: 0x03F

Short explanation:

* For a,b, E = exp – bias with bias is 15.
* For c, negative infinity is 1 11111 000000 and lagest denormalized is 0 00000 111111

3.

1) a

2) c

3) b

4) f

5) g

4.

|  |  |
| --- | --- |
| Destination | Value |
| %edx | 0xC |
| 0x104 | 0x9F |
| 0x10C | 0x110 |
| %eax | 0x10C |
| 0x10C | 0x12 |

5.

int f2(int \*ptr, int x){

int res = x;

for(int i = 0; ptr[i]<=0; i++){

if(res > ptr[i]){

res = ptr[i];

}

}

return res;

}

Short explanation: res is %eax so res = x in the beginning, %ecx is i, %edx is [rdi + 4 \* rdx] equal to ptr[i] and the condition of the loop is ptr[i] <= 0. The condition of if statement is res > ptr[i] and the assignment is res = ptr[i] when if is true.