CS105 – Computer Systems Spring 2023

Week 3: Signed Integers and Floats

SOLUTION

January 30-February 1, 2023

1. Consider a **5-bit** two's complement representation. Fill in the empty boxes in the following table. Addition and subtraction should be performed based on the rules for 5-bit, two's complement arithmetic

Number	Decimal Representation	Binary Representation
n/a	9	`01001`
n/a	-14	`10010`
n/a	12	`01100`
n/a	-12	`10100`
TMax	15	`01111`
TMin	-16	`10000`
TMin+TMin	0	`00000`
TMin+1	-15	`10001`
TMax+1	-16	`10000`
$-$TMax	-15	`10001`
$-$TMin	-16	`10000`

2. The following procedure takes a single-precision floating point number in IEEE format and prints out information about what category of number it is. Fill in the missing code so that it performs this classification correctly.

```
void classify_float(float f){
     /* Unsigned value u has same binary representation as f */
     unsigned u = *(unsigned *) &f;

     /* Split u into the different parts */
     unsigned sign = (u >> 31);      // The sign bit
     unsigned exp  = (u >> 23) & 0xFF;      // The exponent field
     unsigned frac = u & 0x7FFFFF;      // The fraction field

     /* The remaining expressions can be written in terms of the
    values of sign, exp, and frac */

     if (exp == 0 && frac == 0){
         printf("Plus or minus zeron");

     } else if (exp == 0){
         printf("Nonzero, denormalized\n");

     } else if (exp == 0xFF && frac == 0){
         printf("Plus or minus infinity\n");

     } else if (exp == 0xFF){
         printf("NaN\n");

     } else if (exp < 127){
         printf("Greater than -1.0 and less than 1.0\n");

     } else if (sign == 1){
         printf("Less than or equal to -1.0\n");

     } else
         printf("Greater than or equal to 1.0\n");
     }
}
```