

**1**

a)

$$\begin{aligned}
 (-128)_{10} &= (-10000000)_2 \\
 \left(\frac{5}{8}\right)_{10} &= (0.101)_2 \\
 -10000000.101 &= -1.0000000101 * 2^7 \\
 (-134)_{10} &= (-10000110)_2 \\
 (-128.625)_{10} &= (-1)^1 * 2^{(134-127)} * 1.0000000101 = 1 \ 10000110 \ 000000010100000000000000
 \end{aligned}$$

b)

$$\begin{aligned}
 0 \ 00000000 \ 10000000000000000000000000000000 &= (-1)^0 * 2^{-126} * 0.1 \\
 \text{underflow} & \\
 &= 2^{-126} * \frac{1}{2} \\
 &= 5.8774717 * 10^{-39}
 \end{aligned}$$

c)

$$\begin{aligned}
 0 \ 01111111 \ 000000000000000000000000000000 &= (-1)^0 * 2^{(127-127)} * 1.0 \\
 &= 1
 \end{aligned}$$

d)

$$\begin{aligned}
 X &= 1100000111001010011110000000000000 \\
 Y &= 0001111100100000000000000000000000 \\
 -Y &= 1011111001100000000000000000000000 \\
 &= 1100000110000000110000000000000000 \\
 X + (-Y) &= 1100000111001011001110000000000000 \\
 &= -1.00101100111 * 2^4 \\
 &= -18.8046875
 \end{aligned}$$

## 2

```
MOV R2, R0          // R0 = R2
ADD #4, R4, R4       // R4 = R4 + 4
NOP                 // Waiting for R0
NOP                 // Waiting for R0
ADD R0, R2, R1       // R1 = R0 + R2
MOV R4, R2           // R2 = R4
MOV R4, R6           // R6 = MEMORY[R4]
NOP                 // Waiting for R1
MOV R3, R1           // MEMORY[R1] = R3
ADD R0, R2, R3       // R3 = R0 + R2
ADD R4, R6, R5       // R5 = R4 + R6
ADD R2, R4, R1       // R1 = R2 + R4
```

## 3

```
#include <stdio.h>
#include "threads.h"
#define NUMELEMENTS 100
#define NUMPROCESSORS 4

double result = 0,
double x[NUMELEMENTS],
double y[NUMELEMENTS];
volatile int thread_id_counter;

void ConFunc(){
    int i, my_id, start, end;
    double value = 0.0;
    my_id = get_my_thread_id();
    start = my_id * (NUMELEMENTS / NUMPROCESSORS);
    end = (my_id + 1) * (NUMELEMENTS / NUMPROCESSORS) - 1;
    for (i = start; i <= end; i++)
        value += a[i] * b[i];
    while (thread_id_counter != my_id);
    dot_product += value;
    thread_id_counter++;
}

void main(){
    int i;
    for (i = 0; i < NUMELEMENTS; i++){
        x[i] = 0;
        y[i] = 0;
    }
    for (i = 0; i < NUMPROCESSORS-1; i++)
        create_thread(ConFunc);
    ConFunc();
    while(thread_id_counter != NUMPROCESSORS);
    printf("Dot product: %f\n", result);
}
```