Assignment on the Happens-Before Memory Model (HMM)

1. For each program below, tell whether the behavior in question is allowed under the happens-before memory model (HMM), and explain why.

(a)

```
Initially, x = 0.

r1 = x; || r2 = x;
x = 1;  || x = 2;
```

Behavior in question: `r1 = 2, r2 = 1.`

(b)

```
Initially, x = y = 0.

x = 1; || r1 = x; || r3 = y;
y = 1; || r2 = y; || r4 = x;
```

Behavior in question: `r1 = 1, r2 = 0, r3 = 1, r4 = 0.`

(c)

```
Initially, x = y = 0.

r1 = x;                 || r3 = y;
r2 = 1 + r1 * r1 - r1;  || x = r3;
y = r2;                 ||
```

Behavior in question: `r1 = r2 = 1.`

(d)

```
Initially, x = y = z = 0.

r1 = x;          || do {
if (r1 == 0)     ||     r2 = y;
    y = 1;       ||     r3 = z;
else             || } while (r2 + r3 == 0);
    z = 1;       || x = 1;
```

Behavior in question: `r1 = r3 = 1, r2 = 0.`

(e)

```
Initially, x = y = z = w = 0.

r1 = z; || r4 = w;
w = r1; || r3 = y;
r2 = x; || z = r3;
y = r2; || x = 1;
```

Behavior in question: `r1 = r2 = r3 = r4 = 1.`

2. Given two programs C_1 and C_2, and two memory models M_1 and M_2, we write $(C_1, M_1) \sqsubseteq (C_2, M_2)$ to say that all the behaviors generated by C_1 under the memory model M_1 can also be generated by C_2 under the memory model M_2. Here a behavior denotes the values of the registers `r1`, `r2`, ... when the program terminates. For instance, suppose the initial value of `x` is 0, then (`r1 = x`, SC) $\sqsubseteq$ (`r1 = 0`, HMM) holds. This is because `r1 = x` under the SC memory model can generate only one behavior `r1 = 0`, which can also be generated by `r1 = 0` under the happens-before memory model (HMM).

   In each question below, we will give you C_1, C_2, M_1 and M_2. Your job is to analyze whether $(C_1, M_1) \sqsubseteq (C_2, M_2)$ holds. If it holds, just say yes. Otherwise, please describe a behavior that can be generated by C_1 under M_1 but cannot be generated by C_2 under M_2, and explain why.

   (a) C_1 is the program shown on the left side, C_2 is the program shown on the right side, and both M_1 and M_2 are HMM.

```
Initially, x = 0.
x = 1;  ||  x = 2;
r1 = x; ||  r2 = x;
```

```
Initially, x = 0.
r1 = x; ||  r2 = x;
x = 1;  ||  x = 2;
```

   (b) Both C_1 and C_2 are the program shown below, M_1 is HMM, and M_2 is SC.

```
Initially, x = y = 0.
x = 1; ||  x = 2; ||  while (y < 2) {}
y = x; ||  y = x; ||  r1 = x;
```

   (c) C_1 is the program shown on the left side, C_2 is the program shown on the right side, M_1 is HMM, and M_2 is SC.

```
Initially, x = y = 0.
r1 = x;          ||  r2 = y;
if (r1 >= 0)     ||  x = r2;
    y = 1;       ||
```

```
Initially, x = y = 0.
y = 1;  ||  r2 = y;
r1 = x; ||  x = r2;
```

   (d) C_1 is the program shown on the left side, C_2 is the program shown on the right side, M_1 is HMM, and M_2 is SC.

```
Initially, x = y = 0.
r1 = x;          ||  y = 1;
r2 = 42;         ||  x = 1;
if (r1 == 1)     ||
    r2 = y;      ||
```

```
Initially, x = y = 0.
r1 = x;          ||  x = 1;
r2 = 42;         ||  y = 1;
if (r1 == 1)     ||
    r2 = y;      ||
```

   (e) C_1 is the program shown on the left side, C_2 is the program shown on the right side, M_1 is HMM, and M_2 is SC.

```
Initially, x = y = 0.
r1 = x;          ||  r3 = y;
r2 = x;          ||  x = r3;
if (r1 == r2)    ||
    y = 1;       ||
```

```
Initially, x = y = 0.
r1 = x;  ||  r3 = y;
r2 = r1; ||  x = r3;
y = 1;   ||
```