1. In the space below, fill in the body of `div_by_pow_2` so that it returns \( x/2^n \). It should use only bit operations, no floating-point arithmetic. You may not use bit fields (if you happen to know about them). You don’t have to check for error conditions.

```c
union {
    float f;
    uint32_t i;
} b;

uint32_t get_exponent(float x) {
    b.f = x;
    uint32_t mask = 0xff;
    uint32_t expon = (b.i >> 23) & mask;
    return expon;
}

float div_by_pow_2(float x, int n) {
```
2. In ExercisesInC/exercises/ex05 you should find line.c, which includes the following type definition:

```c
typedef struct {
    Point *start, *end;
} Line;
```

Based on this definition, write a function called `make_line` that takes two pointers to `Point` as parameters and returns a pointer to a freshly-allocated `Line`. It should not allocate any new `Point` objects.

3. Draw a stack diagram that shows the state of `main` and `upper_right_rect` at the instant before `upper_right_rect` completes.