This homework is individual. Total points is 15.

1. (4 points) Consider the following C code snippet. The scope of a variable is a range (of lines) where the variable can be accessed. For example, the scope of `mole` is line 2 to line 26. A span is the number of lines between two references of a variable. A reference can be a `def` (e.g., being on the left hand side of an assignment, `a := 0`) or a `use` (e.g., the value is accessed, `a := b` where `b` is used). The live time is the number of lines between the first reference and the last reference.

```
1  int main(void) {
2    double mole=0.0, temp=0.0;
3    int init_vol=0, final_vol=0, inc=0;
4
5    printf("Quantity of carbon dioxide (moles)> ");
6    scanf("%lf", &mole);
7    printf("Temperature (kelvin)> ");
8    scanf("%lf", &temp);
9    printf("Initial volume (milliliters)> ");
10   scanf("%d", &init_vol);
11   printf("Final volume (milliliters)> ");
12   scanf("%d", &final_vol);
13   printf("Volume increment (milliliters)> ");
14   scanf("%d", &inc);
15
16   printf("\n%.6f moles of carbon dioxide at %.1f K\n", mole, temp);
17   printf("Volume (l)\t\tPressure (atm)\n\n");
18
19   int V = init_vol;
20
21   while (V <= final_vol) {
22     printf("%-6f\t\t\t\t%.4f\n", V * L_PER_MILLI,
23                pressure(mole, temp, L_PER_MILLI * V));
24     V += inc;
25   }
26 }
```

Complete the following table:

variable	live time (lines)	span (average) (lines)	scope (line range)
mole	22	6	2,26
init_vol	17	7	3,26
final_vol	19	8	3,26
V	6	0.25	19,26
inc	22	9.5	3,26

2. (3 points) Rewrite/rearrange the code to improve the readability (minimize live time, span, scope). (Also consider using a for-loop.)

```
int main(void) {
   printf("Quantity of carbon dioxide (moles)> ");
   double mole = 0.0;
   scanf("%lf", &mole);
   printf("Temperature (kelvin)> ")
   double temp = 0.0;
   scanf("%lf", &temp);
   printf("Inital volume (milliliters)> ")
   int init_vol = 0;
   scanf("%d", &init_vol);
   printf("Final volume (milliliters)> ")
   int final_vol = 0;
   scanf("%d", &final_vol);
   printf("Volume increment (milliliters)> ")
   int inc = 0;
   scanf("%d", &inc);

   printf("\n%.6f moles of carbon dioxide at %.1f K\n", mole,
temp);
   printf("\nVolume (l)\t\tPressure (atm)\n\n");

   for (int V = init_vol; V <= final_vol; V += inc) {
      printf("%-6f\t\t\t\t%.4f\n", V * L_PER_MILLI,
         pressure(mole, temp, L_PER_MILLI * V));
   }
}
```

3. (4 points) Complete the following table for the updated program:

variable	live time (lines)	span (average) (lines)	scope (line range)
mole	21	5.67	3,24
init_vol	13	5	9,24
final_vol	10	3.5	12,24
V	1	0	21,23
inc	7	2	15,24

4. (2 points) In class, I said Java gives default values to instance variables and requires the programmer to initialize the local variables. What is the C strategy?

C has default values for external and static variables, but local variables get whatever is laying around in memory.

5. (2 points) What is the life span for local variables (in a function/method)?

The life span of local variables is from declaration till the end of the function/method they reside in. Once outside of the function, the variable no longer exists.