Graded out of 34 points.

General observations:

Most people did well in this assignment.

In part 1a, note that it asks for only odd numbers, so 100 should not be printed. Similarly in part 1b, if you start from 100 or 93, 0 should not be printed. For part 1d, some people did not consider the case where x=y. This happens if you only used two if statements (if x>y and if x<y).

For part 3, some people started with an unrealistic space for one person. Also note that your answers should not be overly precise.

Problem 1: $[4 + 4 + 2 + 2]$

(a) Odd numbers from 1 to 100:

```python
for i in range(1,100,2):
    print(i)
```
or any variant that works.

(b) Counting backwards by 7.

```python
i = 100
while i > 0:
    print(i)
    i = i - 7
```

No penalty if you started at 100; the question wasn’t clear. Make sure it doesn’t print a negative value.

(c) It’s computing the minimum of $x$ and $y$.

(d) Any variation of this. No penalties for incorrect syntax as long as it’s clear.

```python
if x > y:
    print(y)
else:
    print(x)
```

Problem 2: [3 + 2 + 1]

(a) In what year will Unix times overflow?

2038. $2^{31} / 60 / 60 / 24 / 365$ converts seconds to 68 years

(b) If the time is unsigned?

2106. Just twice as long as the above.

(c) This might cause overflow (33 bits needed if time is signed). Apparently it actually did cause some programs to break.

Problem 3: [4 + 4 + 4 + 2 + 2]

(a) Dimensions of a square of 10,000 people?

200 feet on a side, or 40,000 sq ft? If people are 2 feet apart, which is maybe too close, that’s reasonable, but maybe 3 ft == 1 meter is better. 1 sq ft per person is a bit unrealistic. -1 for more unrealistic assumptions.

(b) What fraction of the football field?
A football field is 150 ft x 300 ft = 45,000 sq ft, so about 90%. Adjust to taste. The fraction should not have more than about 1 significant digit of precision.

(c) Dimensions of a square of 100 million people?

100M people is 10^4 x 10^4, so 2 x 10,000 = 20,000 feet on a side or 400M sq ft. Again, no excess precision.

-1 for too much precision in 3a, b, c

(d) Half a million is 19 bits (a million is 20 – that's all the arithmetic needed)

(e) 3 bytes