Instructions

- This assignment is due on Tuesday, September 29, 2020 at 2:00 PM EDT. Late submissions will **not** be accepted.
- This assignment consists of one problem with two parts. You must submit both parts to receive full credit.
- Your solution needs to be formatted using the LaTeX template available on OWL. Note that there are different templates available for regular assignments and group assignments. You should use the one for group assignments.
- All group members are expected to be working on the solution and every member should attend all group meetings.
- The Scribe will be submitting the assignment on behalf of the group. It is assumed that every member of the group has proofread the submission.
- All solutions must be written in full sentences.
- You are not allowed to use online resources and should only discuss the solution with members of your group.
- This assignment is worth 5 points.

Part 1.

In this problem, we will be solving quadratic equations modulo prime powers. To this end, we fix an odd prime p and an integer b such that $p \nmid b$. Furthermore, we assume that the congruence

$$X^2 \equiv b \mod p$$

has a solution and we fix one such solution $X = c_1$.

Show that for any positive integer a, the equation

$$X^2 \equiv b \mod p^a$$

also has a solution, say $X = d$, such that $d \equiv c_1 \mod p$.

Hint. Proceed by induction with respect to a. Given a solution modulo p^a, say $X = c_a$, build a solution c_{a+1} modulo p^{a+1} by writing $c_{a+1} = c_a + xp^a$ and solving for x modulo p^{a+1}.

Part 2.

1. Write a function in Python3 called `solve` that, given a prime p, an integer a, and numbers $0 < c, b < p^a$ such that $c^2 \equiv b \mod p$, returns a number d such that $d^2 \equiv b \mod p^a$ and $d \equiv c \mod p$.

2. Download the file `generate_input.py` from OWL, use it to obtain three tuples (p, a, c, b) by running

   ```
   python generate_input.py [last three digits of your student number]
   ```

   and run your program on these three inputs. Here, we use the last three digits of the Programmer's student number.

As part of your submission, please include:

1. the *Python code* implementing your solution;

2. and the three *inputs you generated*, and the *output of your program* run on these three inputs.

Examples

Here are some examples of what your function `solve` should do:

```
>>> solve(3,2,1,4)
 7
>>> solve(5,3,12,19)
 12
>>> solve(3,4,1,4)
 79
```

Notes

- The numbers generated by `generate_input.py` are quite big, so a brute-force solution will not work.
- The file `generate_input.py` is written in Python3, and so should be your solution. Make sure you are using a 64bit version of Python3
- Your code should not make use of any external libraries such as `numpy` or `math`. All the auxiliary functions should be implemented by you, and should be included in your submission. You should only use the most basic arithmetic operations such as `+`, `-`, `*`, `//`, `%`.
- Comments in the code are not mandatory. However in the case of an incorrect solution, the comments can provide grounds for partial credit.