You can form a project group of 2 to 3 persons to solve the problem in project.

Project 1 - Modulo System of Integers

(Credit: The following questions come from the School of Computing)

The descriptions are based on the compiler gcc and the program runs on the UNIX system. You can run your program on the PC system.

Modulo System of Integers

Topic Coverage

- Assignment and expressions
- Control statements
- Functions and procedures

A system of integers modulo `n`, written `Z/n`, is a finite algebraic system consisting of the symbols `0, 1, 2, .., n-1`. In this system, multiplication is defined by the rule "multiply a and b, then divide by n and take the remainder". Using `n = 11` as an example, `2 × 9 ≡ 7 mod 11` (*read as two times nine is equivalent to seven in the modulo eleven system*).

Task

In this task, you are required to write a program that takes an integer as input, and use it to build different tables of values in the modulo system.

Take note of the following:

- Only one sample test case is provided to test for format correctness.
- You should device your own test cases to test your program.

This task is divided into several levels. **Read through all the levels (from first to last, then from last to first) to see how the different levels are related. You may start from any level.**

Level 1

Name your program `mod1.c`

Write a program that takes as input an integer `n (n > 1)` and outputs the value that is read.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed

by a newline character.

```
$ ./a.out
Enter n: 8
8
```

```
$ ./a.out
Enter n: 11
11
```

To proceed to the next level (say level 2), copy your program by typing the Unix command:

```
cp mod1.c mod2.c
```

Level 2

Name your program `mod2.c`

Write a program that takes as input an integer `n (n > 1)` and outputs the multiplication table of the modulo `n` system.

Note that for `n = 11`, there are ten rows and ten columns. The value associated with the first column and first row gives the result of `1 × 1 ≡ 1 mod 11`; the fifth column and fifth row gives the value of `5 × 5 ≡ 3 mod 11`.

Use `%3d` to output each value.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
  1  2  3  4  5  6  7
  2  4  6  0  2  4  6
  3  6  1  4  7  2  5
  4  0  4  0  4  0  4
  5  2  7  4  1  6  3
  6  4  2  0  6  4  2
  7  6  5  4  3  2  1
```

```
$ ./a.out
Enter n: 11
Output Multiplication Table
  1  2  3  4  5  6  7  8  9 10
  2  4  6  8 10  1  3  5  7  9
  3  6  9  1  4  7 10  2  5  8
  4  8  1  5  9  2  6 10  3  7
  5 10  4  9  3  8  2  7  1  6
  6  1  7  2  8  3  9  4 10  5
  7  3 10  6  2  9  5  1  8  4
  8  5  2 10  7  4  1  9  6  3
  9  7  5  3  1 10  8  6  4  2
 10  9  8  7  6  5  4  3  2  1
```

To proceed to the next level (say level 3), copy your program by typing the Unix command:

```
cp mod2.c mod3.c
```

Level 3

Name your program `mod3.c`

Write a program that takes as input an integer `n (n > 1)` and outputs the multiplication table of the modulo `n` system (see previous level).

In addition, output the multiplication table with the header row and leading column.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
---+--------------------
   |  1  2  3  4  5  6  7
---+--------------------
  1|  1  2  3  4  5  6  7
  2|  2  4  6  0  2  4  6
  3|  3  6  1  4  7  2  5
  4|  4  0  4  0  4  0  4
  5|  5  2  7  4  1  6  3
  6|  6  4  2  0  6  4  2
  7|  7  6  5  4  3  2  1
---+--------------------
```

```
$ ./a.out
Enter n: 11
Output Multiplication Table
---+-----------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+-----------------------------
  1|  1  2  3  4  5  6  7  8  9 10
  2|  2  4  6  8 10  1  3  5  7  9
  3|  3  6  9  1  4  7 10  2  5  8
  4|  4  8  1  5  9  2  6 10  3  7
  5|  5 10  4  9  3  8  2  7  1  6
  6|  6  1  7  2  8  3  9  4 10  5
  7|  7  3 10  6  2  9  5  1  8  4
  8|  8  5  2 10  7  4  1  9  6  3
  9|  9  7  5  3  1 10  8  6  4  2
 10| 10  9  8  7  6  5  4  3  2  1
---+-----------------------------
```

To proceed to the next level (say level 4), copy your program by typing the Unix command:

```
cp mod3.c mod4.c
```

Level 4

Name your program `mod4.c`

Write a program that takes as input an integer `n (n > 1)` and outputs two tables:

- the multiplication table of the modulo `n` system (see previous level)
- the multiplicative inverse table of the modulo `n` system

As an example, `b` is a multiplicative inverse of `a` modulo `11` if $a \times b \equiv 1 \bmod 11$.

Thus the multiplicative inverses for `n = 11` are:

- `a = 1; b = 1`
- `a = 2; b = 6`
- `a = 3; b = 4`
- `a = 4; b = 3`
- `a = 5; b = 9`
- `a = 6; b = 2`
- `a = 7; b = 8`
- `a = 8; b = 7`
- `a = 9; b = 5`
- `a = 10; b = 10`

Rather than listing these values, we use the multiplication table, but denote these locations with `*` instead. Note that each column is three-spaces wide.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
---+---------------------
   |  1  2  3  4  5  6  7
---+---------------------
  1|  1  2  3  4  5  6  7
  2|  2  4  6  0  2  4  6
  3|  3  6  1  4  7  2  5
  4|  4  0  4  0  4  0  4
  5|  5  2  7  4  1  6  3
  6|  6  4  2  0  6  4  2
  7|  7  6  5  4  3  2  1
---+---------------------
Output Inverse Table
---+---------------------
   |  1  2  3  4  5  6  7
---+---------------------
  1|  *
  2|
  3|        *
  4|
  5|              *
  6|
  7|                    *
---+---------------------
```

```
$ ./a.out
```

```
Enter n: 11
Output Multiplication Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  1  2  3  4  5  6  7  8  9 10
  2|  2  4  6  8 10  1  3  5  7  9
  3|  3  6  9  1  4  7 10  2  5  8
  4|  4  8  1  5  9  2  6 10  3  7
  5|  5 10  4  9  3  8  2  7  1  6
  6|  6  1  7  2  8  3  9  4 10  5
  7|  7  3 10  6  2  9  5  1  8  4
  8|  8  5  2 10  7  4  1  9  6  3
  9|  9  7  5  3  1 10  8  6  4  2
 10| 10  9  8  7  6  5  4  3  2  1
---+------------------------------
Output Inverse Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  *
  2|                 *
  3|           *
  4|        *
  5|                          *
  6|     *
  7|                       *
  8|                    *
  9|              *
 10|                             *
---+------------------------------
```

To proceed to the next level (say level 5), copy your program by typing the Unix command:

```
cp mod4.c mod5.c
```

Level 5

Name your program `mod5.c`

Write a program that takes as input an integer `n (n > 1)` and outputs three tables:

- the multiplication table of the modulo `n` system (see previous level)
- the multiplicative inverse table of the modulo `n` system (see previous level)
- the list of quadratic residue modulo `n`

Quadratic residues are squares mod `n`, i.e. `a` is a quadratic residue if there is a number `b` such that $b^2 \equiv a \bmod n$.

Using `n = 11` as an example,

- `1` is a quadratic residue since `b = 1, b = 10` will satisfy the above relation
- `3` is a quadratic residue since `b = 5, b = 6` will satisfy the above relation
- `4` is a quadratic residue since `b = 2, b = 9` will satisfy the above relation
- `5` is a quadratic residue since `b = 4, b = 7` will satisfy the above relation

- `9` is a quadratic residue since `b = 3, b = 8` will satisfy the above relation

Use `%3d` to output each value in the list.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
---+---------------------
   |  1  2  3  4  5  6  7
---+---------------------
  1|  1  2  3  4  5  6  7
  2|  2  4  6  0  2  4  6
  3|  3  6  1  4  7  2  5
  4|  4  0  4  0  4  0  4
  5|  5  2  7  4  1  6  3
  6|  6  4  2  0  6  4  2
  7|  7  6  5  4  3  2  1
---+---------------------
Output Inverse Table
---+---------------------
   |  1  2  3  4  5  6  7
---+---------------------
  1|  *
  2|
  3|        *
  4|
  5|              *
  6|
  7|                    *
---+---------------------
Output Quadratic Residue
---+---------------------
  1|  1  3  5  7
  4|  2  6
---+---------------------
```

```
$ ./a.out
Enter n: 11
Output Multiplication Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  1  2  3  4  5  6  7  8  9 10
  2|  2  4  6  8 10  1  3  5  7  9
  3|  3  6  9  1  4  7 10  2  5  8
  4|  4  8  1  5  9  2  6 10  3  7
  5|  5 10  4  9  3  8  2  7  1  6
  6|  6  1  7  2  8  3  9  4 10  5
  7|  7  3 10  6  2  9  5  1  8  4
  8|  8  5  2 10  7  4  1  9  6  3
  9|  9  7  5  3  1 10  8  6  4  2
 10| 10  9  8  7  6  5  4  3  2  1
---+------------------------------
Output Inverse Table
---+------------------------------
```

```
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  *
  2|                 *
  3|           *
  4|        *
  5|                          *
  6|     *
  7|                       *
  8|                    *
  9|              *
 10|                             *
---+------------------------------
Output Quadratic Residue
---+------------------------------
  1|  1 10
  3|  5  6
  4|  2  9
  5|  4  7
  9|  3  8
---+------------------------------
```

To proceed to the next level (say level 6), copy your program by typing the Unix command:

```
cp mod5.c mod6.c
```

Level 6

Name your program `mod6.c`

Write a program that takes as input an integer `n (n > 1)` and outputs three tables and one figure:

- the multiplication table of the modulo `n` system (see previous level)
- the multiplicative inverse table of the modulo `n` system (see previous level)
- the list of quadratic residue modulo `n` (see previous level)
- the fractal of quadratic residues

Note that for the `n = 11` system, `a` is a quadratic residue mod `11` for values of `a = 1, 3, 4, 5, 9`.

For each value of `n = 1, 2, ..., 11`, by marking all quadratic residues mod `n`, a fractal can be constructed.

```
@@@@@@@@@@

 ....@....

  ..@....@

   @@@@@@@

    ....@@

     ...@.

      .@..
```

```
        ...
         @@
          .
```

Columns in the above fractal spans `n = 1` to `n = 11`. For each column, we need to only check if `a = 1, 2, ...,`

`n - 1` is a quadratic residue. Here are some further observations:

- In the leftmost column where `n = 1`, no quadratic residue needs to be checked.
- In the next column where `n = 2`, `a = 1` is a quadratic residue mod `2`.
- In the next column where `n = 3`, `a = 1` is a quadratic residue mod `3`, but `a = 2` is not.
- ...
- In the last column where `n = 11`, `a = 1, 3, 4, 5, 9` are quadratic residue mod `11`,

  while `a = 2, 6, 7, 8, 10` are not.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed

by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
---+--------------------
   |  1  2  3  4  5  6  7
---+--------------------
  1|  1  2  3  4  5  6  7
  2|  2  4  6  0  2  4  6
  3|  3  6  1  4  7  2  5
  4|  4  0  4  0  4  0  4
  5|  5  2  7  4  1  6  3
  6|  6  4  2  0  6  4  2
  7|  7  6  5  4  3  2  1
---+--------------------
Output Inverse Table
---+--------------------
   |  1  2  3  4  5  6  7
---+--------------------
  1|  *
  2|
  3|        *
  4|
  5|              *
  6|
  7|                    *
---+--------------------
Output Quadratic Residue
---+--------------------
  1|  1  3  5  7
  4|  2  6
---+--------------------
Output Fractal
 @@@@@@@
```

```
  ....@.
   ..@..
    @@@@
     ...
      ..
       .
```

```
$ ./a.out
Enter n: 11
Output Multiplication Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  1  2  3  4  5  6  7  8  9 10
  2|  2  4  6  8 10  1  3  5  7  9
  3|  3  6  9  1  4  7 10  2  5  8
  4|  4  8  1  5  9  2  6 10  3  7
  5|  5 10  4  9  3  8  2  7  1  6
  6|  6  1  7  2  8  3  9  4 10  5
  7|  7  3 10  6  2  9  5  1  8  4
  8|  8  5  2 10  7  4  1  9  6  3
  9|  9  7  5  3  1 10  8  6  4  2
 10| 10  9  8  7  6  5  4  3  2  1
---+------------------------------
Output Inverse Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  *
  2|                 *
  3|           *
  4|        *
  5|                          *
  6|     *
  7|                       *
  8|                    *
  9|              *
 10|                             *
---+------------------------------
Output Quadratic Residue
---+------------------------------
  1|  1 10
  3|  5  6
  4|  2  9
  5|  4  7
  9|  3  8
---+------------------------------
Output Fractal
 @@@@@@@@@@
  ....@....
   ..@....@
    @@@@@@@
     ....@@
      ...@.
       .@..
        ...
         @@
          .
```

To proceed to the next level (say level 7), copy your program by typing the Unix command:

```
cp mod6.c mod7.c
```

Level 7

Name your program `mod7.c`

Write a program that takes as input an integer `n (n > 1)` and outputs three tables and one figure:

- the multiplication table of the modulo `n` system (see previous level)
- the multiplicative inverse table of the modulo `n` system (see previous level)
- the list of quadratic residue modulo `n` (see previous level)
- the fractal of quadratic residues (see previous level)

The difference between this level and the preceding one is in the way the fractal is flipped in the output, together with the header and leading indexes.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
Enter n: 8
Output Multiplication Table
---+--------------------
   |  1  2  3  4  5  6  7
---+--------------------
  1|  1  2  3  4  5  6  7
  2|  2  4  6  0  2  4  6
  3|  3  6  1  4  7  2  5
  4|  4  0  4  0  4  0  4
  5|  5  2  7  4  1  6  3
  6|  6  4  2  0  6  4  2
  7|  7  6  5  4  3  2  1
---+--------------------
Output Inverse Table
---+--------------------
   |  1  2  3  4  5  6  7
---+--------------------
  1|  *
  2|
  3|        *
  4|
  5|              *
  6|
  7|                    *
---+--------------------
Output Quadratic Residue
---+--------------------
  1|  1  3  5  7
  4|  2  6
---+--------------------
```

```
Output Fractal
   00000000
   12345678
 08
 07       .
 06      ..
 05     ...
 04    @@@@
 03   ..@..
 02  ....@.
 01 @@@@@@@
```

```
$ ./a.out
Enter n: 11
Output Multiplication Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  1  2  3  4  5  6  7  8  9 10
  2|  2  4  6  8 10  1  3  5  7  9
  3|  3  6  9  1  4  7 10  2  5  8
  4|  4  8  1  5  9  2  6 10  3  7
  5|  5 10  4  9  3  8  2  7  1  6
  6|  6  1  7  2  8  3  9  4 10  5
  7|  7  3 10  6  2  9  5  1  8  4
  8|  8  5  2 10  7  4  1  9  6  3
  9|  9  7  5  3  1 10  8  6  4  2
 10| 10  9  8  7  6  5  4  3  2  1
---+------------------------------
Output Inverse Table
---+------------------------------
   |  1  2  3  4  5  6  7  8  9 10
---+------------------------------
  1|  *
  2|                 *
  3|           *
  4|        *
  5|                          *
  6|     *
  7|                       *
  8|                    *
  9|              *
 10|                             *
---+------------------------------
Output Quadratic Residue
---+------------------------------
  1|  1 10
  3|  5  6
  4|  2  9
  5|  4  7
  9|  3  8
---+------------------------------
Output Fractal
   00000000011
   12345678901
 11
 10          .
 09         @@
 08        ...
 07       .@..
```

```
 06      ...@.
 05     ....@@
 04    @@@@@@@
 03   ..@....@
 02  ....@....
 01 @@@@@@@@@@
```

```
$ ./a.out
Enter n: 23
Output Multiplication Table
---+-----------------------------------------------------------------------
   |  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
---+-----------------------------------------------------------------------
  1|  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
  2|  2  4  6  8 10 12 14 16 18 20 22  1  3  5  7  9 11 13 15 17 19 21
  3|  3  6  9 12 15 18 21  1  4  7 10 13 16 19 22  2  5  8 11 14 17 20
  4|  4  8 12 16 20  1  5  9 13 17 21  2  6 10 14 18 22  3  7 11 15 19
  5|  5 10 15 20  2  7 12 17 22  4  9 14 19  1  6 11 16 21  3  8 13 18
  6|  6 12 18  1  7 13 19  2  8 14 20  3  9 15 21  4 10 16 22  5 11 17
  7|  7 14 21  5 12 19  3 10 17  1  8 15 22  6 13 20  4 11 18  2  9 16
  8|  8 16  1  9 17  2 10 18  3 11 19  4 12 20  5 13 21  6 14 22  7 15
  9|  9 18  4 13 22  8 17  3 12 21  7 16  2 11 20  6 15  1 10 19  5 14
 10| 10 20  7 17  4 14  1 11 21  8 18  5 15  2 12 22  9 19  6 16  3 13
 11| 11 22 10 21  9 20  8 19  7 18  6 17  5 16  4 15  3 14  2 13  1 12
 12| 12  1 13  2 14  3 15  4 16  5 17  6 18  7 19  8 20  9 21 10 22 11
 13| 13  3 16  6 19  9 22 12  2 15  5 18  8 21 11  1 14  4 17  7 20 10
 14| 14  5 19 10  1 15  6 20 11  2 16  7 21 12  3 17  8 22 13  4 18  9
 15| 15  7 22 14  6 21 13  5 20 12  4 19 11  3 18 10  2 17  9  1 16  8
 16| 16  9  2 18 11  4 20 13  6 22 15  8  1 17 10  3 19 12  5 21 14  7
 17| 17 11  5 22 16 10  4 21 15  9  3 20 14  8  2 19 13  7  1 18 12  6
 18| 18 13  8  3 21 16 11  6  1 19 14  9  4 22 17 12  7  2 20 15 10  5
 19| 19 15 11  7  3 22 18 14 10  6  2 21 17 13  9  5  1 20 16 12  8  4
 20| 20 17 14 11  8  5  2 22 19 16 13 10  7  4  1 21 18 15 12  9  6  3
 21| 21 19 17 15 13 11  9  7  5  3  1 22 20 18 16 14 12 10  8  6  4  2
 22| 22 21 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
---+-----------------------------------------------------------------------
Output Inverse Table
---+-----------------------------------------------------------------------
   |  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
---+-----------------------------------------------------------------------
  1|  *
  2|                                    *
  3|                        *
  4|                  *
  5|                                          *
  6|           *
  7|                              *
  8|        *
  9|                                                      *
 10|                     *
 11|                                                               *
 12|     *
 13|                                                *
 14|              *
 15|                                                         *
 16|                                       *
 17|                                                   *
 18|                           *
 19|                                                *
 20|                                          *
```

```
 21|                                *
 22|                                                                    *
---+--------------------------------------------------------------------
Output Quadratic Residue
---+--------------------------------------------------------------------
  1|  1 22
  2|  5 18
  3|  7 16
  4|  2 21
  6| 11 12
  8| 10 13
  9|  3 20
 12|  9 14
 13|  6 17
 16|  4 19
 18|  8 15
---+--------------------------------------------------------------------
Output Fractal
   00000000011111111112222
   12345678901234567890123
 23
 22                       .
 21                      ..
 20                     .@.
 19                    ....
 18                   ..@.@
 17                  .@....
 16                 @@@@@@@
 15                .@...@@.
 14               .......@.
 13              ...@@....@
 12             @........@@
 11            ..@....@..@.
 10           ..@.@..@.....
 09          @@@@@@@@@@@@@@@
 08         .....@..@.....@
 07       .@....@...@@.@..
 06      ...@....@...@...@
 05     ....@@.......@@.@.
 04    @@@@@@@@@@@@@@@@@@@
 03   ..@....@.@........@@
 02  ....@......@..@.....@
 01 @@@@@@@@@@@@@@@@@@@@@@@
```