MATH 495R, HOMEWORK 8
MULTIPLICATIVE GROUPS MODULO N

1. Modular Exponentiation

1. Any positive integer has a binary representation; i.e. a representation as a sum of distinct powers of two (see the math 290 textbook, Proposition 15.3).

   We represent a number as a sum of distinct powers of two by writing it in binary. The binary representation of a number consists of a string of 0's and 1's, for instance 10111001. In this string the digits tell whether a specific power of two is or is not included in the sum representing the number. The rightmost digit represents 2^0, the second digit from the right represents 2^1, and so on. So the binary number above represents the number

$$\begin{aligned} 1\cdot 2^7+0\cdot 2^6+1\cdot 2^5+1\cdot 2^4+1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+1\cdot 2^0 &= 2^7+2^5+2^4+2^3+2^0\\ &= 128+32+16+8+1\\ &= 185. \end{aligned}$$

   Write a function `binary(n)` that will convert an integer `n` into its binary representation (as a string). You should not use any commands built in to python that automatically convert integers to binary.

   Test your function on the following numbers:

   The binary representation of 73 is 1001001.
   The binary representation of 965 is 1111000101.
   The binary representation of 14352654 is 110110110000000100001110.
   The binary representation of 2093984 is 111111111001110100000.

2. Write a function `power(a,b,n)` that will compute $\overline{a}^b$ in $\mathbb{Z}_n$. Have it return the answer as an integer k with $0 \le k < n$. It should work for very large values of a, b and n. It should **not** use the built in python function `pow(a,b,n)`.

   Use it to compute $\overline{2}^{1234567890}$ in $\mathbb{Z}_n$ for $n = 1234567891$

   Hints: To compute $a^{(2^n)}$, we can compute $(((a^2)^2)^{\cdots})^2$ with n two's in the exponent.

   In computing a^{2^n}, we could square a to get $a_2 = a^2$. Then $a_4 = a_2^2 = a^4$, and $a_8 = a_4^2 = a^8$, and so on. If we want to compute this value in $\mathbb{Z}_n$, we can reduce it modulo n after each squaring, so that we never need to work with extremely large numbers.

   To compute an arbitrary power of a, we will combine binary notation with the ability to compute $a^{(2^n)}$. So, for instance, to compute a^{73}, we use the fact that

$$73 = 1\cdot 2^6+0\cdot 2^5+0\cdot 2^4+1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+1\cdot 2^0 = 2^6+2^3+2^0,$$

   and compute

$$a^{(2^6)}\cdot a^{(2^3)}\cdot a^{(2^0)}.$$

As an example, if we want to compute 6^{73} in $\mathbb{Z}_{11}$, we could compute

$$\begin{aligned} 6^2 &\equiv 36 \equiv 3 \pmod{11} \\ 6^4 &\equiv (6^2)^2 \equiv 3^2 \equiv 9 \pmod{11} \\ 6^8 &\equiv (6^4)^2 \equiv 9^2 \equiv 81 \equiv 4 \pmod{11} \\ 6^{16} &\equiv (6^8)^2 \equiv 4^2 \equiv 16 \equiv 5 \pmod{11} \\ 6^{32} &\equiv (6^{16})^2 \equiv 5^2 \equiv 25 \equiv 3 \pmod{11} \\ 6^{64} &\equiv (6^{32})^2 \equiv 3^2 \equiv 9 \pmod{11} \end{aligned}$$

Then, using the binary representation of 73 as $2^6 + 2^3 + 2^0 = 64 + 8 + 1$, we find

$$6^{73} \equiv 6^{64}6^8 6^1 \equiv 9 \cdot 4 \cdot 6 \equiv 9 \cdot 24 \equiv 9 \cdot 2 \equiv 18 \equiv 7 \pmod{11}.$$

Note that this method uses only 8 multiplications mod 11 to compute 6^{73} mod 11.

3. Recall that the set

$$\mathbb{Z}_n^\times = \{\bar{a} : 1 \le a \le n, \gcd(a, n) = 1\}$$

is a group. The Euler φ function evaluated at n is defined to be the size of this group.

Write a function `eulerphi(n)` that determines the size of $\mathbb{Z}_n^\times$. Test your code by finding $\left|\mathbb{Z}_{11}^\times\right| = 10$, $\left|\mathbb{Z}_{56}^\times\right| = 24$, and $\left|\mathbb{Z}_{120}^\times\right| = 32$.

4. Write a function `order(a,n)` that returns the order of a in the multiplicative group $\mathbb{Z}_n^\times$. If a is not relatively prime to n, it should return the string "Number not relatively prime to modulus". Test your function by finding $o(\bar{4}) = 5$ in $\mathbb{Z}_{11}^\times$, $o(\overline{33}) = 6$ in $\mathbb{Z}_{56}^\times$, $o(\bar{7}) = 4$ in $\mathbb{Z}_{120}^\times$

5. For each number n between 2 and 100, determine the size of $\mathbb{Z}_n^\times$, and determine whether the group $\mathbb{Z}_n^\times$ is cyclic. If the group is cyclic, determine a generator. Print your results in the following format:

   2 1
   3 2
   4 3
   5 2
   6 5
   7 3
   8
   ⋮

   where the line for n is "n g", where g is a generator of $\mathbb{Z}_n^\times$ if $\mathbb{Z}_n^\times$ is cyclic, and just "n" if $\mathbb{Z}_n^\times$ is not cyclic.