1　やったこと

RSA 暗号を実装した。数字を暗号にし、復号することができる。

2　仕組み

素数 p、q を十分大きくとっておく。e を、$(p-1)(q-1)$ との最大公約数が 1 となるようにとる。これを公開する。つぎに $r=pq$、$L=(p-1)(q-1)$ と r、L を定めておく。e は $\gcd(e,L)=1$ となるようにとっていたので、

$$^{\exists}d,k\in\mathbb{Z}\ \text{ s.t. }\ de-kL=1 \tag{1}$$

となる d,k がとれる。また、

$$x^L\equiv 1 \pmod{r} \tag{2}$$

なので、

$$\begin{aligned} x^{ed} &\equiv x^{1+kL} \pmod{r} \\ &\equiv x\times 1^k \pmod{r} \\ &\equiv x \pmod{r} \end{aligned}$$

となり、暗号化されて受信した数 x^e を d 乗すると暗号化する前の x が復元できる。ただし上の (1) は、整数環は PID であることより e と L で生成されたイデアルの和は、ある a で生成されたイデアルに一致し、a は e、L をともにわりきるので $a=1$ とわかるのでよい。また (2) はラグランジュの定理と中国式剰余定理よりわかる。

3　プログラムにする上で注意したこと

n 乗する計算は普通に書くと $O(n)$ になってしまうが、指数の偶奇をうまくわけることで $O(\log n)$ になるようにした。

$$x^n=\begin{cases} x^{n/2} & (n:even) \\ x^{n-1}\cdot x & (n:odd) \end{cases}$$

これを再帰を使って書いている。また z_syz（不定方程式の解を見つける関数）の返り値は負の可能性があるので d が負のときは正になるように書いた。