Vector products
α - the angle between a, b.
$\cos\alpha$ = dot product of normalized vectors
dot product $= (a_x \cdot b_x + a_y \cdot b_y)/(|a| \cdot |b|)$.
cross product $= a_x \cdot b_y - a_y \cdot b_x$.

Vector rotation

$$x' = x\cos\alpha - y\sin\alpha$$
$$y' = x\sin\alpha + y\cos\alpha$$

Triangle area
Heron's formula:
$$s = \frac{a+b+c}{2}$$
$$A = \sqrt{s(s-a)(s-b)(s-c)}$$
Heron's formula - stable:
$$a \geq b \geq c$$
$$A = \frac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}$$
Cross product: Two vectors starting from [0,0]:
$$A = \frac{1}{2}|x_1y_2 - x_2y_1|$$

Simple polygon's area
Let $x_n, y_n = x_0, y_0$. Then:
$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_iy_{i+1} - x_{i+1}y_i)$$
The vertices must be ordered counterclockwise. If they are ordered clockwise the area will be negative but correct in absolute value.

Pick's theorem
If a polygon's vertices are grid points:
$$A = i + \frac{1}{2}b - 1$$
A - area of the polygon
i - inner points
b - points on the boundary

Point-line distance
Point: (x_0, y_0)
Line: $[(x_1, y_1), (x_2, y_2)]$
$$d = \frac{|(x_2-x_1)(y_1-y_0) - (x_1-x_0)(y_2-y_1)|}{\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}}$$

Line-line intersection
Line 1: $[(x_1, y_1), (x_2, y_2)]$
Line 2: $[(x_3, y_3), (x_4, y_4)]$
$$x = \frac{(x_1y_2 - y_1x_2)(x_3-x_4) - (x_1-x_2)(x_3y_4 - y_3x_4)}{(x_1-x_2)(y_3-y_4) - (y_1-y_2)(x_3-x_4)}$$
$$y = \frac{(x_1y_2 - y_1x_2)(y_3-y_4) - (y_1-y_2)(x_3y_4 - y_3x_4)}{(x_1-x_2)(y_3-y_4) - (y_1-y_2)(x_3-x_4)}$$

Line-circle intersection
Line: $[(x_1, y_1), (x_2, y_2)]$
Circle: $[(0, 0), r]$
$$d_x = x_2 - x_1 \qquad d_y = y_2 - y_1 \qquad d_r = \sqrt{d_x^2 + d_y^2}$$
$$D = x_1y_2 - x_2y_1 \qquad \Delta = r^2d_r^2 - D^2$$
$\Delta < 0 \implies$ no intersection
$\Delta = 0 \implies$ tangent
$\Delta > 0 \implies$ intersection
$$x = \frac{Dd_y \pm \operatorname{sgn}^*(d_y)d_x\sqrt{\Delta}}{d_r^2}$$
$$y = \frac{-Dd_x \pm |d_y|\sqrt{\Delta}}{d_r^2}$$
where
$$\operatorname{sgn}^*(x) = \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

Circle-circle intersection
$$(x-x_0)^2 + (y-y_0)^2 = r_0^2$$
$$x^2 + y^2 = r_1^2$$
$$d = x_0^2 + y_0^2$$
$$p = \sqrt{((r_0+r_1)^2 - d)\,(d - (r_1-r_0)^2)}$$
$$A_x = \frac{x_0}{2} + \frac{y_0p - x_0\left(r_0^2 - r_1^2\right)}{2d}$$
$$A_y = \frac{y_0}{2} + \frac{-x_0p - y_0\left(r_0^2 - r_1^2\right)}{2d}$$
$$B_x = \frac{x_0}{2} + \frac{-y_0p - x_0\left(r_0^2 - r_1^2\right)}{2d}$$
$$B_y = \frac{y_0}{2} + \frac{x_0p - y_0\left(r_0^2 - r_1^2\right)}{2d}$$

Quadratic equation
$$ax^2 + bx + c = 0, \quad b \neq 0$$
$$t = -\left(b + \operatorname{sgn}(b)\sqrt{b^2 - 4ac}\right)/2$$
$$x_1 = t/a$$
$$x_2 = c/t$$

Pythagorean Triplets
$$a^2 + b^2 = c^2$$
Euclid's generating formula:
$a = v^2 - u^2$
$b = 2uv$
$c = u^2 + v^2$
where u and $v > u$ are relatively prime and of opposite parity. The formula generates a set of distinct triples containing precisely the primitive triples (after appropriately sorting $(v^2 - u^2)$ and $2uv$).

Primes
32003
32009
65003
65011

999 999 929
999 999 937
1 000 000 007
1 000 000 009

2 147 483 549
2 147 483 563

999 999 999 999 999 967
999 999 999 999 999 989
1 000 000 000 000 000 003
1 000 000 000 000 000 009

9 223 372 036 854 775 643
9 223 372 036 854 775 783

Divisors
Let $n = p^a$, where p is a prime.
Number of divisors: $(a+1)$.
Sum of divisors: $(p^{a+1}-1)/(p-1)$.
Both are multiplicative functions.

Euler's totient function
$\varphi(n)$ the number of positive integers less than or equal to n that are coprime to n.

$$\sum_{d|n} \varphi(d) = n$$

$$\varphi(p^k) = (p-1)p^{k-1}$$

Examples: $\varphi(9) = 6$, $\varphi(36) = 12$
It is multiplicative. n and a coprime $\implies a^{\varphi(n)} \equiv 1 \mod n$.

Sums

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2$$

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$$

Probability
Conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Bayes' theorem:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Euler's formula for planar graphs

$$vertices - edges + faces = 2$$

String hashing

```
unsigned long djb2(unsigned char *str)
{
    unsigned long hash = 5381;
    int c;
    while (c = *str++)
        hash = hash*33 + c;
    return hash;
}
```

Gray code
Gray code is a binary numeral system where two successive values differ in only one bit.
`G(n) = n ^ (n >> 1)`

Lexicographically next bit permutation
Example: N = 3: ..., 00010101, 00010110, 00011001, ...

```
unsigned int v; // current permutation of bits
unsigned int w; // next permutation of bits

unsigned int t = v | (v - 1);
w = (t + 1) | (((~t & -~t) - 1)
    >> (__builtin_ctz(v) + 1));
```

GCC hacks
You can append l or ll for long and long long variants.

`int __builtin_ffs(unsigned int x)`
Returns one plus the index of the least significant 1-bit of x, or if x is zero, returns zero.

`int __builtin_clz(unsigned int x)`
Returns the number of leading 0-bits in x, starting at the most significant bit position. If x is 0, the result is undefined.

`int __builtin_ctz(unsigned int x)`
Returns the number of trailing 0-bits in x, starting at the least significant bit position. If x is 0, the result is undefined.

`int __builtin_popcount(unsigned int x)`
Returns the number of 1-bits in x.

`T std::__gcd(T a, T b)`
Returns the greatest common divisor of two integer values.

`T __gnu_cxx::power(T x, int n)`
Returns x^n, where $n >= 0$. Include file: `ext/numeric`.