Exponentiation

Exponentiation is a mathematical operation, written as x^n, involving two numbers, the base x and the exponent or power n. When n is a positive integer, exponentiation corresponds to repeated multiplication of the base: that is, x^n is the product of multiplying n bases: $x^n = x * x * \dots * x$.

How to find this value if x and n are given? We can use just simple loop with complexity O(n) like

```
res = 1;
for (i = 1; i <= n; i++)
  res = res * x;
```

Can we do it faster? For example, $x^{10} = (x^5)^2 = (x * x^4)^2 = (x * (x^2)^2)^2$.
We can notice that $x^{2n} = (x^2)^n$, or $x^{100} = (x^2)^{50}$.

$$x^n = \begin{cases} \left(x^2\right)^{n/2}, n \text{ is even} \\ x \cdot x^{n-1}, n \text{ is odd} \\ 1, n = 0 \end{cases}$$

```
int f(int x, int n)
{
  if (n == 0) return 1;
  if (n % 2 == 0) return f(x * x, n / 2);
  return x * f(x, n - 1);
}
```

Complexity O($\log_2 n$).

E-OLYMP **273. Modular Exponentiaion** Three positive integers x, n and m are given. Find the value of x^n mod m.

► In this problem $2 \le n \le 10^7$, in the case of applying algorithm with O(n) complexity, the program will accept the time limit because you need to make no more than 10^7 multiplications (its no more than 1 second).

E-OLYMP **5198. Modular Exponentiaion** Find the value of x^n mod m.

► Use the function f(x, n, m) = x^n mod m.

```
long long f(long long x, long long n, long long m)
{
  if (n == 0) return 1;
  if (n % 2 == 0) return f((x * x) % m, n / 2, m);
  return (x * f(x, n - 1, m)) % m;
}
```

E-OLYMP 4439. Exponentiaion Find the value of a^b. It is known that the answer does not exceed 10^{18}.

► If you use O(n) complexity algorithm, for the test case 1 ^ (10^{18}) you get ***Time Limit*** because in this case you must make 10^{18} multiplications. Use O($\log_2 n$) complexity algorithm.

E-OLYMP 9644. Sum of powers Find the value of the sum

$$(1^n + 2^n + 2 * 3^n + 3 * 4^n + 4 * 5^n + ... + 99 * 100^n) \bmod m$$

► The first two terms of the sum differ from the rest. Let's calculate them separately. Next, we calculate the sum in a loop of i from 3 to 100, each term has the form $(i - 1) * i^n$.

E-OLYMP 5493. Just Add it For two given integers n and k find the value of

$$(Z_n + Z_{n\text{-}1} - 2Z_{n\text{-}2}) \bmod 10000007,$$

where $Z_n = S_n + P_n$, $S_n = 1^k + 2^k + 3^k + ... + n^k$ and $P_n = 1^1 + 2^2 + 3^3 + ... + n^n$.

► Let's simplify the required expression:

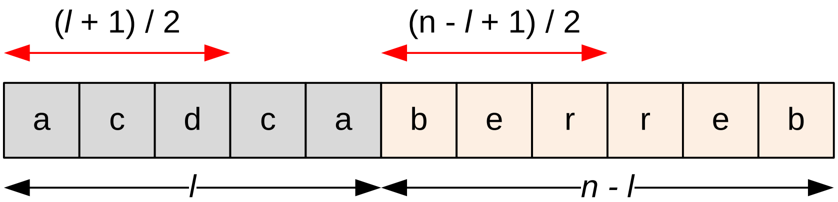
$$Z_n + Z_{n\text{-}1} - 2Z_{n\text{-}2} = S_n + S_{n\text{-}1} - 2 * S_{n\text{-}2} + P_n + P_{n\text{-}1} - 2 * P_{n\text{-}2} =$$
$$(1^k + 2^k + 3^k + ... + n^k) + (1^k + 2^k + 3^k + ... + (n-1)^k) -$$
$$2*(1^k + 2^k + 3^k + ... + (n-2)^k) + (1^1 + 2^2 + 3^3 + ... + n^n) +$$
$$(1^1 + 2^2 + 3^3 + ... + (n-1)^{n\text{-}1}) - 2*(1^1 + 2^2 + 3^3 + ... + (n-2)^{n\text{-}2}) =$$
$$= n^k + 2(n-1)^k + n^n + 2(n-1)^{n\text{-}1}$$

It remains to calculate the sum of the four terms, taken modulo 10000007.

E-OLYMP 9592. Concatenation of two palindromes Find the number of ways to construct a string of length n using k Latin letters (the size of the alphabet is k) in the form of concatenation of two nonempty palindromes.

► We represent a string of length n as the concatenation of two non-empty palindromes of lengths l and $n - l$. In the palindrome of length l, the first $(l + 1) / 2$ letters can be chosen arbitrarily (any of the k letters available), and the remaining letters should be chosen so as to obtain a palindrome. This can be done in $k^{(l+1)/2}$ ways.

Similarly, in the palindrome of length $n - l$, the first $(n - l + 1) / 2$ letters can be chosen arbitrarily, and the rest should form a palindrome. This can be done in $k^{(n\text{-}l+1)/2}$ ways.



We can construct a concatenation of two palindromes with lengths l and $n - l$ in

$$k^{(l+1)/2} * k^{(n\text{-}l+1)/2}$$

ways. Since none of the palindromes can be empty, then $1 \le l \le n - 1$. The total number of possible palindromes equals to

$$\sum_{l=1}^{n-1} k^{\frac{l+1}{2}} \cdot k^{\frac{n-l+1}{2}}$$

All operations should be carried out modulo $10^9 + 7$.

E-OLYMP 9557. Bins and balls There are n bins in a row. There is also an infinite supply of balls of n distinct colors. Place exactly one ball into each bin, with the restriction that adjacent bins cannot contain balls of the same color. How many valid configurations of balls in bins are there?

► Any of n balls can be put into the first box. The color of the ball in the second box must not match the color of the ball in the first box. Therefore, you can put any ball of $n – 1$ colors in the second box. In the i-th box, you can put a ball of any color that does not match the color of the ball in the $(i – 1)$-th box.

1	2	3	4		n
n	n-1	n-1	n-1	...	n-1

Thus, the number of different arrangements of balls in the boxes equals to

$$n * (n – 1)^{n – 1} \bmod 10^9 + 7$$

E-OLYMP 9616. Anti-palindromic strings Two integers n and m are given. Find the number of strings of length n, which symbols belong to the alphabet of size m, that do not contain palindromes of length more than one as substrings.

► If a string does not contain a substring that is a palindrome of length 2 or length 3, then it does not contain a substring that is a palindrome of length greater than one.

If $n = 1$, then a string of length 1 can contain any of m letters. The required number of strings equals to m.

If $m = 1$ and $n > 1$, the answer is 0, since there is only one string *aa..aa* and it contains the palindrome *aa*.

If $n = 2$, then the first position of the string can contain any of m letters, and the second position can contain any of $m – 1$ letters (the letters in the first and second positions must not coincide, forming a palindrome). The number of strings equals to $(m * (m – 1)) \bmod 10^9 + 7$.

Let the length of the input string be greater than 2. Any of m letters can be in the first position, and any of $m – 1$ letters can be in the second position. The third position must be different from the second (the second and third letters must not form a palindrome). The third position must be different from the first (the first three letters must not form a palindrome). This means that any of $m – 2$ letters can be in the third position.

1	2	3	4		n
m	m-1	m-2	m-2	...	m-2

Following this logic, we conclude that the letter at the i-th position must differ from the letters at positions $(i – 1)$ and $(i – 2)$. The number of required string equals to $(m * (m – 1) * (m – 2)^{n – 2}) \bmod 10^9 + 7$.