

<Yourein/ints>

getdivisors

任意の整数 N について、正整数の約数を列挙してvector型で返します。

Definition

```
std::vector<long long> getdivisors(long long n)
```

使用例

```
long long n = 10;

auto v = Yourein::ints::getdivisors(n);

for (auto x : v){
    cout << x << " ";
}
cout << endl;
```

出力結果

```
1 2 5 10
```

注意点 少なくともlong long型で扱える値の範囲は計算できますが、そこまで大きな数だと計算時間が破綻する可能性があります。この関数はエラトステネスの篩を用いた高速約数列挙ではなく、 \sqrt{N} までの試し割りを実際に行う関数であるためです。

よって、この関数の計算時間は $O(\sqrt{N})$ です。

kth_divisor

上の関数の拡張関数で任意の整数 N において k 番目の約数を返します。

Definition

```
long long kth_divisor(long long n, long long kth)
```

使用例

```
long long n = 10;
cout << Yourein::ints::kth_divisor(n, 3) << endl;
```

出力結果

5

注意点 内部的にgetdivisor関数呼んで約数リストを取得してから k 番目の約数を返しています。そのため、計算時間がgetdivisor関数と同等です。

したがって、この関数の計算量は $O(\sqrt{N})$ です。

modPow

繰り返し二乗法を用いて $a^n \bmod p$ を計算します。

Definition

```
long long modPow(long long a, long long n, long long p)
```

繰り返し二乗法を用いるので n^{10^9} などの計算も常識的な計算時間で行うことができます。