

## LÝ THUYẾT SỐ

1.Số Nguyên Tố .....	2
2.Số Chính Phương.....	4
3. Tổng Ước Số Của Số Nguyên .....	5
4. Đếm ước số của một số nguyên .....	6
5. Số Hoàn Hảo.....	7
6. Số Hoàn Hảo Thuật Toán Tốt Hơn.....	8
7. Ước chung lớn nhất, Bội chung nhỏ nhất .....	9
9.Phi Hàm Euler.....	11
10. Tổ hợp 1 .....	12
11. Giải thuật Euclid mở rộng.....	13
12. Nghịch đảo modun.....	15
13.Nghịch Đảo Modun áp dụng Định Lý Nhỏ Fermat .....	16
14. Tính $A^B \% C$ .....	17
15. Tính Đối Xứng .....	18
16. Tổ Hợp 2 Áp Dụng Quy Hoạch Động .....	19
17.Tổ hợp 3 Áp Dụng Nghịch Đảo Modun.....	20
18. Phân Tích Thừa Số Nguyên Tố .....	21
19. Số Fibonacci .....	22
20. Số Fibonacci Chia Dư với $1e9 + 7$ .....	23
21. Số T-Prime : Số có đúng 3 ước.....	24
22. Bậc của thừa số nguyên tố P trong N! : Công thức Legendre .....	25
23. Tam giác Pascal .....	26
24. Sàng số nguyên tố Eratosthenes.....	27
25. Sàng Số Nguyên Tố Trong đoạn [L, R].....	28

# 28Tech - Become A Better Developer

“Trên bước đường thành công không có dấu chân của người lười biếng”

Website : <https://28tech.com.vn>

Youtube : [https://www.youtube.com/28tech\\_](https://www.youtube.com/28tech_)

Facebook cá nhân : <https://www.facebook.com/andrew28042711/>

Group : [www.facebook.com/groups/28techgroup/](https://www.facebook.com/groups/28techgroup/)

28TECH

## 1.Số Nguyên Tố

```
PRIME_NUMBER

#include <bits/stdc++.h>
using namespace std;

int prime(int n){
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            return false;
        }
    }
    return n > 1;
}

int main(){
    int n; cin >> n;
    cout << prime(n) << endl;
}
```

## 2.Số Chính Phương

```
SQUARE_NUMBER

#include <bits/stdc++.h>
using namespace std;

bool squareNumber(long long n){
    int m = sqrt(n);
    if(1ll * m * m == n){
        return true;
    }
    else{
        return false;
    }
}

int main(){
    int n; cin >> n;
    cout << squareNumber(n) << endl;
}
```

### 3. Tổng Ước Số Của Số Nguyên

```
SQUARE_NUMBER

#include <bits/stdc++.h>
using namespace std;

int tongUoc(int n){
    int sumDiv = 0;
    for(int i = 1; i <= sqrt(n); i++){
        if(n % i == 0){
            sumDiv += i;
            if(i != n / i){
                sumDiv += n / i; // ???
            }
        }
    }
    return sumDiv;
}

int main(){
    int n; cin >> n;
    cout << tongUoc(n) << endl;
}
```

#### 4. Đếm ước số của một số nguyên

```
SQUARE_NUMBER

#include <bits/stdc++.h>
using namespace std;

int demUoc(int n){
    int countDiv = 0;
    for(int i = 1; i <= sqrt(n); i++){
        if(n % i == 0){
            countDiv += 1;
            if(i != n / i){
                countDiv += 1; // ???
            }
        }
    }
    return countDiv;
}

int main(){
    int n; cin >> n;
    cout << demUoc(n) << endl;
}
```

## 5. Số Hoàn Hảo

```

SỐ Hoàn Hảo

#include <bits/stdc++.h>
using namespace std;

bool soHoanHao(long long n){
    long long sum = 1;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            sum += i;
            if(i != n / i){
                sum += n / i;
            }
        }
    }
    return sum == n;
}

int main(){
    long long n; cin >> n;
    cout << soHoanHao(n) << endl;
}
```

## 6. Số Hoàn Hảo Thuật Toán Tốt Hơn

```
Số Hoàn Hảo Áp Dụng Định Lý Euler-Euclid

#include <bits/stdc++.h>
using namespace std;

bool prime(long long n){
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0) return false;
    }
    return n > 1;
}

bool soHoanHao(long long n){
    for(int i = 1; i <= 32; i++){
        if(prime(i)){
            long long x = (long long)pow(2, i - 1);
            if(prime(x)){
                long long hh = x * ((long long)pow(2, i) - 1);
                if(hh == n) return true;
            }
        }
    }
    return false;
}

int main(){
    long long n; cin >> n;
    cout << soHoanHao(n) << endl;
}
```



## 7. Ước chung lớn nhất, Bội chung nhỏ nhất

```
UỚC CHUNG LỚN NHẤT - BỘI CHUNG NHỎ NHẤT

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll gcd1(ll a, ll b){
    if(b == 0) return a;
    else return gcd1(b, a % b);
}

ll gcd2(ll a, ll b){
    while(b){
        ll r = a % b;
        a = b;
        b = r;
    }
    return a;
}

ll lcm(ll a, ll b){
    return a / gcd1(a, b) * b;
}

int main(){
    ll x, y; cin >> x >> y;
    cout << gcd1(x, y) << ' ' << lcm(x, y) << endl;
}
```

## 8. Lũy Thừa Nhị Phân

```
TÍNH A^B ÁP DỤNG LŨY THỪA NHỊ PHẦN

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll binpow1(ll a, ll b){
    ll res = 1;
    while(b){
        if(b % 2 == 1){
            res *= a;
        }
        a *= a;
        b /= 2;
    }
    return res;
}

ll binpow2(ll a, ll b){
    if(b == 0) return 1;
    ll tmp = binpow2(a, b / 2);
    if(b % 2 == 1) return tmp * tmp * a;
    else return tmp * tmp;
}

int main(){
    cout << binpow1(2, 10) << ' ' << binpow2(2, 10) << endl;
}
```

## 9. Phi Hàm Euler

```
PHI HÀM EULER

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll phi(ll n){
    ll res = n;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            res -= res / i;
            while(n % i == 0){
                n /= i;
            }
        }
    }
    if(n > 1) res -= res / n;
    return res;
}

int main(){
    cout << phi(17) << endl;
}
```

## 10. Tổ hợp 1

```
TÍNH TỔ HỢP CHẬP K CỦA N

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll toHop(int n, int k){
    k = min(k, n - k);
    ll res = 1;
    for(int i = 1; i <= k; i++){
        res *= (n - i + 1);
        res /= i;
    }
    return res;
}

int main(){
    cout << toHop(10, 2) << endl;
}
```

## 11. Giải thuật Euclid mở rộng

```
● ● ● GIẢI THUẬT EUCLID MỞ RỘNG :  $AX + BY = \text{GCD}(A, B)$ 

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

int extended_gcd(int a, int b, int &x, int &y){
    if(b == 0){
        x = 1; y = 0;
        return a;
    }
    int x1, y1;
    int gcd = extended_gcd(b, a % b, x1, y1);
    x = y1;
    y = x1 - a / b * y1;
    return gcd;
}

int main(){
    //ax + by = gcd(a, b)
    int a = 16, b = 10;
    int x, y;
    int d = extended_gcd(a, b, x, y);
    cout << d << endl; // gcd(a, b) = 2
    cout << x << ' ' << y << endl; // x = 2, y = -3
}
```

28TECH

## 12. Nghịch đảo modun

```
● ● ●          NGHỊCH ĐẢO MODUN

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

int extended_gcd(int a, int b, int &x, int &y){
    if(b == 0){
        x = 1; y = 0;
        return a;
    }
    int x1, y1;
    int gcd = extended_gcd(b, a % b, x1, y1);
    x = y1;
    y = x1 - a / b * y1;
    return gcd;
}

void inv(int a, int b){
    int x, y;
    int gcd = extended_gcd(a, b, x, y);
    if(gcd != 1){
        cout << "Khong ton tai nghich dao modun";
    }
    else{
        cout << (x % b + b) % b << endl;
    }
}

int main(){
    inv(16, 7); // 16 * 4 % 7 = 1
}
```

### 13. Nghịch Đảo Moduln áp dụng Định Lý Nhỏ Fermat

```
● ● ●          NGHỊCH ĐẢO MODULN - ỨNG DỤNG KHI B LÀ SỐ NGUYÊN TỐ

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll powMod(ll a, ll b, ll c){
    ll res = 1;
    while(b){
        if(b % 2){
            res *= a; res %= c;
        }
        a *= a;
        a %= c;
        b /= 2;
    }
    return res;
}

ll inv(ll a, ll b){
    return powMod(a, b - 2, b); // Áp dụng khi b là số nguyên tố
}

int main(){
    cout << inv(16, 7) << endl;
}
```



#### 14. Tính $A^B \% C$

```
TÍNH A ^ B % C

#include <bits/stdc++.h>
using namespace std;

using ll = long long;

ll powMod(ll a, ll b, ll c){
    ll res = 1; a %= c;
    while(b){
        if(b % 2){
            res *= a; res %= c;
        }
        a *= a;
        a %= c;
        b /= 2;
    }
    return res;
}

int main(){
    cout << powMod(2, 1000, 10) << endl; //6
}
```

## 15. Tính Đối Xứng

```
● ● ● SÔ ĐỐI XỨNG

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

bool doiXung(ll n){
    ll reverse = 0, tmp = n;
    while(n){
        reverse = reverse * 10 + n % 10;
        n /= 10;
    }
    return tmp == reverse;
}

int main(){
    cout << doiXung(1234321) << endl;
}
```

## 16. Tổ Hợp 2 Áp Dụng Quy Hoạch Động

```
TÍNH TỔ HỢP CHẬP K CỦA N CHIA DƯ CHO  $10^9 + 7$ 

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

ll C[1001][1001];

const int mod = 1e9 + 7;

void init(){
    for(int i = 0; i <= 1000; i++){
        for(int j = 0; j <= i; j++){
            if(j == 0 || i == j) C[i][j] = 1;
            else C[i][j] = (C[i - 1][j - 1] + C[i - 1][j]) % mod;
        }
    }
}

int main(){
    init();
    int n, k; cin >> n >> k;
    cout << C[n][k] << endl;
}
```

## 17. Tổ hợp 3 Áp Dụng Nghịch Đảo Modul

```
TÍNH TỔ HỢP CHẬP K CỦA N CHIA DƯ CHO  $10^9 + 7$ 

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;
const int mod = 1e9 + 7;
ll fac[1000001];

ll powMod(ll a, ll b, ll c){
    ll res = 1;
    while(b){
        if(b % 2){
            res *= a; res %= c;
        }
        a *= a;
        a %= c;
        b /= 2;
    }
    return res;
}

ll inv(ll a, ll m){
    return powMod(a, m - 2, m);
}

int main(){
    fac[0] = 1;
    for(int i = 1; i <= (int)1e6; i++){
        fac[i] = fac[i - 1] * i; fac[i] %= mod;
    }
    int n, k; cin >> n >> k;
    cout << fac[n] * (inv(fac[k], mod) * inv(fac[n - k], mod) % mod) % mod;
}
```

## 18. Phân Tích Thừa Số Nguyên Tố

```
● ● ● PHÂN TÍCH THỪA SỐ NGUYÊN TỐ

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

void factorize(ll n){
    for(int i = 2; i <= sqrt(n); i++){
        while(n % i == 0){
            cout << i << ' ';
            n /= i;
        }
    }
    if(n > 1) cout << n << endl;
}

int main(){
    factorize(100); // 100 = 2*2*5*5
}
```

● ● ● SỞ FIBONACCI DÙNG ĐỀ QUY - QUY HOẠCH ĐỘNG

```
#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

// Độ phức tạp  $1.618^n$ 

ll fibo(int n){
    if(n <= 1) return n;
    else return fibo(n - 1) + fibo(n - 2);
}

// Độ phức tạp  $O(N)$ 

ll fibo2(int n){
    ll f[n + 1];
    f[0] = 0; f[1] = 1;
    for(int i = 2; i <= n; i++){
        f[i] = f[i - 1] + f[i - 2];
    }
    return f[n];
}

int main(){
    cout << fibo2(92) << endl;
    cout << fibo(92) << endl;
}
```

## 20. Số Fibonacci Chia Dư với $10^9 + 7$

```
● ● ● SÔ FIBONACCI CHIA DƯ VỚI  $10^9 + 7$ 

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

const int mod = 1e9 + 7;

ll fibo(int n){
    ll f[n + 1];
    f[0] = 0; f[1] = 1;
    for(int i = 2; i <= n; i++){
        f[i] = (f[i - 1] + f[i - 2]) % mod;
    }
    return f[n];
}

int main(){
    cout << fibo(100000) << endl;
}
```

## 21. Số T-Prime : Số có đúng 3 ước số

```
● ● ● SÔ T PRIME - SÔ CÓ ĐÚNG 3 ƯỚC SÔ

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer
using ll = long long;

bool prime(ll n){
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0) return false;
    }
    return n > 1;
}

bool t_prime(ll n){
    ll m = sqrt(n);
    if(m * m != n) return false;
    return prime(m);
}

int main(){
    for(int i = 1; i <= 1000; i++){
        if(t_prime(i)) cout << i << ' ';
    }
    //4 9 25 49 121 169 289 361 529 841 961
}
```



## 22. Bậc của thừa số nguyên tố P trong N! : Công thức Legendre

```
● ● ● BẬC CỦA THỪA SỐ NGUYÊN TỐ P TRONG N!  
  
#include <bits/stdc++.h>  
using namespace std;  
28Tech - Become A Better Developer  
using ll = long long;  
  
//Tìm x lớn nhất sao cho n! chia hết cho p^x  
int legendre(int n, int p){  
    int ans = 0;  
    for(int i = p; i <= n; i *= p){  
        ans += n / i;  
    }  
    return ans;  
}  
  
int main(){  
    cout << legendre(10, 5) << endl;  
}
```

### 23. Tam giác Pascal

```
TAM GIÁC PASCAL

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

//Tam giác pascal với h là chiều cao

void pascal(int h){
    ll p[h][h];
    for(int i = 0; i < h; i++){
        for(int j = 0; j <= i; j++){
            if(j == 0 || i == j) p[i][j] = 1;
            else p[i][j] = p[i - 1][j - 1] + p[i - 1][j];
            cout << p[i][j] << ' ';
        }
        cout << endl;
    }
}

int main(){
    pascal(5);
}

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

## 24. Sàng số nguyên tố Eratosthenes

```
● ● ● SÀNG SỐ NGUYÊN TỐ

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

const int maxn = (int)1e7;

int prime[maxn + 1];

void sang(){
    for(int i = 0; i <= maxn; i++){
        prime[i] = 1;
    }
    prime[0] = prime[1] = 0;
    for(int i = 2; i <= sqrt(maxn); i++){
        if(prime[i]){
            for(int j = i * i; j <= maxn; j += i){
                prime[j] = 0;
            }
        }
    }
}

int main(){
    sang(); // sau khi sàng bạn có thể check số nguyên tố trong [1, 1e7]
    for(int i = 1; i <= 20; i++){
        if(prime[i]) cout << i << ' ';
    }
}
Output : 2 3 5 7 11 13 17 19
```

## 25. Sàng Số Nguyên Tố Trong đoạn [L, R]

```
● ● ● SÀNG SỐ NGUYÊN TỐ TRÊN ĐOẠN [L, R]

#include <bits/stdc++.h>
using namespace std;
28Tech - Become A Better Developer

using ll = long long;

void sang(ll l, ll r){
    vector<bool> prime(r - l + 1, true);
    for(ll i = 2; i <= sqrt(r); i++){
        for(ll j = max(i * i, (l + i - 1) / i * i); j <= r; j += i){
            prime[j - l] = false;
        }
    }
    for(ll i = max(l, 2ll); i <= r; i++){
        if(prime[i - l]) cout << i << ' ';
    }
}

int main(){
    sang(30, 50);
}
Output : 31 37 41 43 47
```