

# 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>

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

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

Phần đề quy này bạn nào chưa hiểu rõ cách hoạt động của nó có thể xem chi tiết video hướng dẫn của mình tại :

[https://www.youtube.com/watch?v=eQ3VpTtc9IE&ab\\_channel=28tech](https://www.youtube.com/watch?v=eQ3VpTtc9IE&ab_channel=28tech)

28TECH

## MỤC LỤC

1.Thuật toán tìm kiếm nhị phân .....	3
2.Hàm Lower_bound, hoạt động tương tự như hàm có sẵn trong thư viện STL...	4
3.Hàm Upper_bound trong thư viện STL .....	5
4.Số thập phân sang hệ 16(Hex) .....	6
4.Số thập phân sang hệ nhị phân.....	7
5.Đếm các giá trị riêng biệt trong mảng.....	8
6.In các số khác nhau trong mảng.....	9
7. In các số trong mảng theo chiều thuận, nghịch.....	11
8. Mảng đối xứng .....	12
9.Số lớn nhất, nhỏ nhất trong mảng .....	14
10. In số theo chiều thuận, nghịch .....	15
11.Đếm số chữ số của một số nguyên dương .....	16
12.Giai thừa.....	17
13. Lũy thừa nhị phân ( Rất quan trọng).....	19
14.Số fibonacci.....	21
15.Tổ hợp .....	22
16.Tổng chữ số của một số nguyên dương .....	23
17.Ước chung lớn nhất.....	24

## 1. Thuật toán tìm kiếm nhị phân

```
TÌM KIẾM NHỊ PHÂN

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

Tìm kiếm nhị phân

bool binary_search(int a[], int left, int right, int x){
    if(left > right){
        return false;
    }
    else{
        int mid = (left + right) / 2;
        if(a[mid] == x){
            return true;
        }
        else if(a[mid] < x)
            return binary_search(a, mid + 1, right, x);
        else
            return binary_search(a, left, mid - 1, x);
    }
}

int main(){
    int n = 8;
    int a[] = {1, 2, 3, 4, 5, 6, 8, 10};
    int x = 2;
    cout << binary_search(a, 0, n - 1, x) << endl;
}
Output : 1
```

## 2. Hàm Lower\_bound, hoạt động tương tự như hàm có sẵn trong thư viện STL

```
LOWER_BOUND

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

Lower_bound : Tìm chỉ số của phần tử đầu tiên >= x trong mảng a[] có n phần tử
Nếu không tìm thấy phần tử thỏa mãn trả về n

int lower_bound(int a[], int left, int right, int x){
    if(left > right){
        return right + 1;
    }
    else{
        int mid = (left + right) / 2;
        if(a[mid] == x){
            return min(mid, lower_bound(a, left, mid - 1, x));
        }
        else if(a[mid] < x)
            return lower_bound(a, mid + 1, right, x);
        else
            return lower_bound(a, left, mid - 1, x);
    }
}

int main(){
    int n = 8;
    int a[] = {1, 2, 2, 2, 2, 6, 8, 10};
    int x = 2;
    cout << lower_bound(a, 0, n - 1, x) << endl;
}

Output : 1
```

### 3. Hàm Upper\_bound trong thư viện STL

```
UPPER_BOUND

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

Upper_bound : Tìm chỉ số của phần tử đầu tiên > x trong mảng a[] có n phần tử
Nếu không tìm thấy phần tử thỏa mãn trả về n

int upper_bound(int a[], int left, int right, int x){
    if(left > right){
        return right + 1;
    }
    else{
        int mid = (left + right) / 2;
        if(a[mid] > x){
            return min(mid, upper_bound(a, left, mid - 1, x));
        }
        else
            return upper_bound(a, mid + 1, right, x);
    }
}

int main(){
    int n = 8;
    int a[] = {1, 2, 2, 2, 2, 6, 8, 10};
    int x = 2;
    cout << upper_bound(a, 0, n - 1, x) << endl;
}
Output : 5
```

#### 4.Số thập phân sang hệ 16(Hex)

```
● ● ●          CHUYỂN SỐ THẬP PHÂN SANG HỆ 16

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

Chuyển số thập phân sang hệ 16
using ll = long long;

void DecimalToHex(ll n){
    if(n <= 15){
        if(n < 10) cout << n;
        else cout << (char)(n + 55);
        return;
    }
    else{
        DecimalToHex(n / 16);
        ll r = n % 16;
        if(r < 10) cout << r;
        else cout << (char)(r + 55);
    }
}

int main(){
    int n = 9892;
    DecimalToHex(n);
}

Output : 26A4
```

#### 4.Số thập phân sang hệ nhị phân

```
● ● ●          CHUYỂN SỐ THẬP PHÂN SANG NHỊ PHÂN

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

Chuyển số thập phân sang nhị phân
using ll = long long;

void DecimalToBinary(ll n){
    if(n <= 1){
        cout << n;
        return;
    }
    else{
        DecimalToBinary(n / 2);
        cout << n % 2;
    }
}

int main(){
    int n = 9892;
    DecimalToBinary(n);
}

Output : 10011010100100
```

## 5. Đếm các giá trị riêng biệt trong mảng



ĐẾM CÁC SỐ KHÁC NHAU TRONG MẢNG

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

Đếm các số khác nhau trong mảng theo thứ tự xuất hiện

```
int count(int a[], int n){
    if(n == 0){
        return 0;
    }
    else{
        bool found = false;
        for(int i = n - 2; i >= 0; i--){
            if(a[i] == a[n - 1]){
                found = true; break;
            }
        }
        if(found) return count(a, n - 1);
        else return 1 + count(a, n - 1);
    }
}

int main(){
    int a[] = {1, 2, 6, 4, 5, 1, 2, 2};
    cout << count(a, 8) << endl;
}
```

Output : 5



## 6. In các số khác nhau trong mảng

```
IN CÁC SỐ KHÁC NHAU TRONG MẢNG

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

In ra các số khác nhau trong mảng theo thứ tự xuất hiện

void in(int a[], int n){
    if(n == 0){
        return;
    }
    else{
        bool found = false;
        for(int i = n - 2; i >= 0; i--){
            if(a[i] == a[n - 1]){
                found = true; break;
            }
        }
        in(a, n - 1);
        if(!found) cout << a[n - 1] << ' ';
    }
}

int main(){
    int a[] = {1, 2, 6, 4, 5, 1, 2, 2};
    in(a, 8);
}

Output :
1 2 6 4 5
```

28TECH

## 7. In các số trong mảng theo chiều thuận, nghịch

IN MẢNG THEO CHIỀU THUẬN, NGHỊCH

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

using ll = long long;

void leftToRight(int a[], int n){
    if(n == 0){
        return; //?????
    }
    leftToRight(a, n - 1);
    cout << a[n - 1] << ' ';
}

void rightToLeft(int a[], int n){
    if(n == 0) return;
    cout << a[n - 1] << ' ';
    rightToLeft(a, n - 1);
}

int main(){
    int a[] = {1, 2, 3, 4, 5};
    leftToRight(a, 5);
    cout << endl;
    rightToLeft(a, 5);
}
```

Output :

```
1 2 3 4 5
5 4 3 2 1
```

## 8. Mạng đối xứng

28TECH

MẢNG ĐỐI XỨNG ( PALINDROME)

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

28Tech - Become A Better Developer

using ll = long long;

bool doiXung(int a[], int l, int r){
    if(l >= r)
        return true;
    if(a[l] != a[r])
        return false;
    else
        return doiXung(a, l + 1, r - 1);
}

int main(){
    int n, a[1000];
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];
    cout << doiXung(a, 0, n - 1);
}

Input :
5
1 2 3 2 1
Output : 1
```

## 9. Số lớn nhất, nhỏ nhất trong mảng

```
● ● ● SỞ LỚN NHẤT, NHỎ NHẤT TRONG MẢNG

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

28Tech - Become A Better Developer

using ll = long long;

int maxEle(int a[], int n){
    if(n == 1){
        return a[0];
    }
    else{
        return max(a[n - 1], maxEle(a, n - 1));
    }
}

int minEle(int a[], int n){
    if(n == 1){
        return a[0];
    }
    else{
        return min(a[n - 1], minEle(a, n - 1));
    }
}

int main(){
    int n, a[1000];
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];
    cout << maxEle(a, n) << ' ' << minEle(a, n) << endl;
}

Input :
5
1 2 3 2 1
Output : 3 1
```

## 10. In số theo chiều thuận, nghịch

IN CHỮ SỐ THEO CHIỀU THUẬN, NGHỊCH

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

using ll = long long;

void thuan(ll n){
    if(n < 10){
        cout << n << ' ';
        return; // ???
    }
    else{
        thuan(n / 10);
        cout << n % 10 << ' ';
    }
}

void nghich(ll n){
    if(n < 10){
        cout << n << endl;
        return;
    }
    cout << n % 10 << ' ';
    nghich(n / 10);
}

int main(){
    ll n; cin >> n;
    thuan(n);
    cout << endl;
    nghich(n);
}

Input : 12345
Output :
1 2 3 4 5
5 4 3 2 1
```

## 11. Đếm số chữ số của một số nguyên dương

● ● ● ĐỀM SỐ CHỮ SỐ CỦA SỐ NGUYÊN DƯƠNG

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

28Tech - Become A Better Developer


using ll = long long;

int countDigit(int n){
    if(n < 10){
        return 1;
    }
    else{
        return 1 + countDigit(n / 10);
    }
}

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



## 12. Giai thừa



```
● ● ● GIAI THỪA

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

28Tech - Become A Better Developer

using ll = long long;

ll giaiThua(int n){
    if(n == 0) return 1;
    else return n * giaiThua(n - 1);
}

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

28TECH

### 13. Lũy thừa nhị phân ( Rất quan trọng)

```
LŨY THỪA VÀ LŨY THỪA CHIA DƯ

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

28Tech - Become A Better Developer

using ll = long long;

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

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

int main(){
    cout << binpow(2, 10) << endl;
    cout << powMod(2, 1000000, 10) << endl;
}

Output :
1024
6
```

28TECH

## 14.Số fibonacci

```

SỐ FIBONACCI

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

28Tech - Become A Better Developer

using ll = long long;

ll fibo(int n){
    if(n == 0 || n == 1)
        return n;
    else{
        return fibo(n - 1) + fibo(n - 2);
    }
}

ll F[100];
//Đệ quy có nhớ
ll fibo2(int n){
    if(F[n]) return F[n];
    if(n <= 1){
        return n;
    }
    else{
        F[n] = fibo2(n - 1) + fibo2(n - 2);
        return F[n];
    }
}

int main(){
    int n; cin >> n;
    cout << fibo2(92) << endl;
}
Output : 7540113804746346429
```

## 15. Tổ hợp

```
Tính Tổ Hợp Dùng Đệ Quy

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

using ll = long long;

int nCk(int n, int k){
    //base case
    if(k == 0 || k == n){
        return 1;
    }
    else{
        return nCk(n - 1, k - 1) + nCk(n - 1, k);
    }
}

int main(){
    int n, k; cin >> n >> k;
    cout << nCk(n, k) << endl;
}
```

## 16. Tổng chữ số của một số nguyên dương

```
● ● ● TÔNG CHỮ SỐ CỦA SỐ NGUYÊN DƯƠNG

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

28Tech - Become A Better Developer

using ll = long long;

int sumDigit(int n){
    if(n < 10){
        return n;
    }
    else{
        return n % 10 + sumDigit(n / 10);
    }
}

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

## 17. Ước chung lớn nhất

```
UỚC CHUNG LỚN NHẤT

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

28Tech - Become A Better Developer

using ll = long long;

int gcd(int a, int b){
    if(b == 0) return a;
    else return gcd(b, a % b); // gcd(a, b) = gcd(b, a % b)
}

int main(){
    int a, b; cin >> a >> b;
    cout << gcd(a, b) << endl;
}
```



28TECH