

以下无先后顺序

常用函数

1. 闰年判断

```
int leapyear(int year){
    return (year % 4 == 0 && year % 100 != 0 || year % 400 == 0);
}
```

2. 交换swap(a,b)

```
#define swap(a b){\
    __typeof(a) temp = a;\
    a = b; b = temp;\
}
```

注意亦或方式虽然可以交换ab但是前提是a!=b;

c++直接使用swap即可

3. 最大公约数最小公倍数

```
int gcd(int a, int b){
    return b == 0 ? a : gcd(b, a%b);
}
```

注意：默认a>b

```
int lcm(int a, int b){
    return a * b / gcd(a, b);
}
```

注意：默认a>b

4. 约数个数

```
//o(n)
int factoriale(int x){
    int num = 0;
    for(int i = 1; i <= x; i++){
        if(x % i) continue;
        num++;
    }
    return num;
}
//o(根号n)
```

```

int factoriale(int x){
    int num = 0, i = 1;
    while(i * i <= x){
        if(x % i) continue;
        num += 2;
    }
    if(i * i == x) num--;
    return num;
}
//比上一个更低，就对了
int factoriale(int x){
    int num = 1;
    for(int i = 2; i * i <= x; i++) {
        if(x % i) continue;
        int times = 0;
        while(x % i == 0) x /= i, times++;
        num *= (times + 1);
    }
    if(x != 1) num *= 2;
    return num;
}

```

5.max

```

#define MAX(a,b){\
    __typeof(a) aa = (a);\
    __typeof(b) bb = (b);\
    aa > bb ? aa : bb ;\
}

```

c++一般直接使用max函数即可

6.阶乘

```

int fac(int n){
    return n == 1 ? 1 : fac(n - 1) * n;
}

```

7.素数判断

```

int is_prime(int x){
    if(x <= 1) return 0;
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0) return 0;
    }
    return 1;
}

```

8.回文数判断

整形

```
int is_palindrome(int x){
    int m = x, n = 0;
    while(m) {
        n = n * 10 + m % 10;
        m /= 10;
    }
    return n == x;
}
```

字符串版本

```
bool isPalindrome(int x) {
    if (x < 0) return 0;
    char s[20];
    sprintf(s, "%d", x);
    for (int i = strlen(s) - 1, j = 0; j < i; i--, j++) {
        if (s[i] != s[j]) return 0;
    }
    return 1;
}
```

9.10进制到n进制转换

```
void func(long long n, int x) {
    stack<int> st;
    while (n != 0) {
        st.push(n % x);
        n = n / x;
    }
    while (!st.empty()) {
        cout << sta.top();
        sta.pop();
    }
    cout << endl;
}
```

注意上面只是2-9

10.整数翻转

```
int reverse(int x) {
    int sum = 0,temp = 0;
    while(x) {
        temp = x % 10;
        x /= 10;
        if (sum > INT_MAX / 10 || (sum == INT_MAX / 10 && temp > 7)) return 0;
        if (sum < INT_MIN / 10 || (sum == INT_MIN / 10 && temp < -8)) return 0;
        sum = sum * 10 + temp;
    }
    return sum;
}
```

一般很少有人卡边界，所以if两行既麻烦也不好记，一般都用long long方便

11.素数筛

12.二分

13.快速幂

```
int poww(int a,int b){
    int ans = 1;
    while(b){
        if(b & 1) ans *= a;
        a *= a;
        b >>= 1;
    }
    return ans;
}
```

一般还是用long long型比较好

14.矩阵乘法

```

int a[x][y],int b[y][z];
int c[x][z];
int func() {
    for(int i = 1;i <= x; i++){
        for(int j = 1; j <= z; j++){
            c[i][j] = 0;
            for(int k = 1; k <= y; k++){
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}

```

这里为代码逻辑展示，不代表可以编译通过

15.矩阵的阶乘

15.两点距离

16.回文串长度

```

int func(string &str,int l,int r) {
    int cnt = 0;
    while(l >= 0 && r <= str.size()) {
        if (str[l] != str[r]) break;
        l--,r++;
        cnt += 2;
    }
    return cnt;
}
//调用
int x = func(str,i - 1,i + 1) + 1;
int y = func(str,i,i + 1);

```

17.大整数加减乘除

常用技巧

set

set的妙用主要是根据，set的性质，set可以按顺序输出，set内部元素有序，但是set可以和pair进行配合，又会有很多用途，具体问题具体分析吧

map

map跟set相同，更加的灵活

输入

整行输入处理:

###