1.

```c
#include<stdio.h>

int main() {
    int sum3, sum5, sum15;
    sum3 = (3 + 999) * 333 / 2;
    sum5 = (5 + 995) * 199 / 2;
    sum15 = (15 + (1000 / 15) * 15) * (1000 / 15) / 2;
    printf("%d\n", sum3 + sum5 - sum15);
    return 0;
}
```

2.

```c
#include<stdio.h>

int main() {
    int i, a[200000] = {0};
    a[1] = 1;
    a[2] = 1;
    int s = 0;
    for(i = 3; a[i - 1] <= 4000000; i++) {
        a[i] = a[i - 1] + a[i - 2];
        if(a[i] % 2 == 0) {
            s += a[i];
        }
    }
    printf("%d", s);
    return 0;
}
```

3.

```c
#include <stdio.h>
#include <inttypes.h>
#define NUM 600851475143

int main() {
    int64_t num = NUM, n = 2, ans = 0;
    while (n * n <= num) {
        if(num % n == 0) ans = n;
        while(num % n == 0) num /= n;
        n += 1;
    }
    if (num != 1) ans = num;
    printf("%"PRId64"\n", ans);
    return 0;
}
```

4.

```c
#include<stdio.h>

int hui(int x) {
    int s = x, h = 0;
    while (s) {
        h *= 10;
        h += s % 10;
        s /= 10;
    }
    if(h == x) {
        return 1;
    }
    return 0;
}

int main() {
    int i, j, max = 0, s;
    for(i = 100; i < 1000; i++) {
        for(j = i; j < 1000; j++) {
            s = i * j;
            if(hui(s) == 1) {
                if(s > max) max = s;
            }
        }
    }
    printf("%d", max);
}
```

5.

```c
#include<stdio.h>

int yue(int x, int y) {
    return y == 0 ? x : yue(y, x % y);
}

int main() {
    int s = 1;
    for(int i = 1; i <= 20; i++) {
        if(s % i == 0) continue;
        s = s / yue(s, i) * i;
    }
    printf("%d", s);
}
```

6.

```c
#include<stdio.h>
#include<inttypes.h>
#define MAX_N 100

int main() {
    int64_t s = 0, sum = 0, h;
    for(int i = 1; i <= 100; i++) {
        s += i;
        sum += i * i;
    }
    h = sum - s * s;
    printf("%"PRId64"\n", h);
}
```

7.

```c
#include<stdio.h>

#define MAX_N 2000000
int a[MAX_N + 5] = {0};

int main() {
    for(int i = 2; i <= MAX_N; i++) {
        if (!a[i]) {
            a[++a[0]] = i;
        }
        for (int j = 1; j <= a[0]; j++) {
            if (a[j] * i >= MAX_N) break;
            a[a[j] * i] = 1;
            if(i % a[j] == 0) break;
        }
    }
    printf("%d", a[10001]);
}
```

8.

```c
#include<stdio.h>
#include<inttypes.h>

int main() {
    while(scanf("%c") != EOF) {

    }
    int64_t p = 1, zero = 0;
    for (int i = 0; num[i]; i++) {
        if (num[i] == '0') {
            zero += 1;
        } else {
            p *= (num[i] - '0');
        }
        if (i >= 13) {
            if (num[i - 13] == '0') {
                zero -= 1;
            } else {
                p /= (num[i - 13] - '0');
            }
        }
        if (zero == 0 && p > ans) ans = p;
    }
    printf("%"PRId64 "\n", ans);
    return 0;
}
```

9.

```
#include<stdio.h>

int main() {
    int a, b, c, s;
    for (int m = 32; m >= 1; m--) {
        for(int n = 1; n < m; n++) {
            a = 2 * n * m;
            b = m * m - n * n;
            c = m * m + n * n;
            if(a + b + c == 1000) {
                s = a * b * c;
                printf("%d\n", s);
                return 0;
            }
        }
    }
}
```

10.

```
#include<stdio.h>
#include<inttypes.h>
#define MAX_N 2000000

int prime[MAX_N + 5] = {0};

int main() {
    int64_t sum = 0, i, j;
    for( i = 2; i <= MAX_N; i++) {
        if (!prime[i]) prime[++prime[0]] = i;
        for(j = 1; j <= prime[0]; j++) {
            if(prime[j] * i >= MAX_N) break;
            prime[prime[j] * i] = 1;
            if(i % prime[j] == 0) break;
        }
    }
    for(i = 1; i < prime[0]; i++) {
        sum += prime[i];
    }
    printf("%"PRId64"\n", sum);
}
```

11.

```
#include<stdio.h>

int a[30][30] = {0};
int w[4][2] = {0, 1, 1 , 1, 1 , 0, 1, -1};

int main() {
    int x, y, m, n, max = 0, s, p;
    for(int i = 5; i < 25; i++) {
        for(int j = 5; j < 25; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    for(x = 5; x < 25; x++) {
        for(y = 5; y < 25; y++) {
            for(int i = 0; i < 4; i++) {
                s = 1, m = x, n = y;
                for(int j = 0; j < 4; j++) {
                    s *= a[m][n];
                    m += w[i][0];
                    n += w[i][1];
                }
                if (s > max) {
                    max = s;
                    printf("%d\n", s);
                }
            }
        }
    }
    printf("%d\n", max);
}
```

13.大加

```
#include<stdio.h>
#include<string.h>
char num[55];
int ans[55] = {1, 0};
int main() {
    for(int i = 0; i < 100; i++) {
        scanf("%s", num);
        int len = strlen(num);
        if (ans[0] < len) ans[0] = len;
        for (int j = 0; j < len; j++) {
            ans[len - j] += (num[j] - '0');
        }
        for (int j = 1; j <= ans[0]; j++) {
            if(ans[j] < 10) continue;
            ans[j + 1] += ans[j] / 10;
            ans[j] %= 10;
            ans[0] += (j == ans[0]);
        }
    }
    for(int i = ans[0]; i > ans[0]  - 10; i--) {
        printf("%d", ans[i]);
    }
    printf("\n");
    return 0;
}
```

14.

```
#include<stdio.h>
#include<inttypes.h>
#define MAX_N 6000000

int a[MAX_N] = {0};

int64_t aaa(int64_t x) {
    if(x == 1) return 1;
    if(x < 6000000 && a[x] != 0) return a[x];
    int64_t i;
    if(x % 2 == 0) return i = aaa(x / 2) + 1;
    if(x % 2 != 0) return i = aaa(3 * x + 1) + 1;
}
int main() {
    int64_t s = 0, i, h,ans;
    for (int i = 1; i < 1000000; i++) {
        h = aaa(i);
        if(h > s) {
            s = h;
            ans = i;
        }
    }
    printf("%"PRId64"\n", ans);
}
```

15.

```
#include<stdio.h>
#include<inttypes.h>

int main() {
    int64_t s = 1, n = 40, w = 1;
    while (n > 20) {
        s *= n;
        if(s % n == 0) s /= w;
        n--;
        if(w >= 20) {
            w = 1;
            continue;
        }
        w++;
    }
    printf("%" PRId64 "\n", s);
    return 0;
}
```

16.大数乘法

```
#include<stdio.h>

int main() {
    int num[4000] = {1, 1};
    for (int i = 0; i < 50; i++) {
        for (int j = 1; j <= num[0]; j++) {
            num[j] *= 1024 * 1024;
        }
        for (int j = 1; j <= num[0]; j++) {
            if (num[j] < 10) continue;
            num[j + 1] += num[j] / 10;
            num[j] %= 10;
            num[0] += (j == num[0]);
        }
    }
    int ans = 0;
    for (int i = 1; i <= num[0]; i++) {
        ans += num[i];
    }
    printf("%d\n", ans);
    return 0;
}
```

17.

```
#include<stdio.h>

int main() {
    int s;
    int aa[8] = {6, 6, 6, 5, 5, 7, 6, 5};//20 ～ 70的头
    int a[19] = {3, 3, 5, 4, 4, 3, 5, 5, 4, 3, 6, 6, 8, 8, 7, 7, 9, 8, 8};//前19个
    int b[9] = {0};
    int s9 = 36;
    int s19 = 0;
    for(int i = 0; i < 19; i++) {
        s19 += a[i];
    }
    int s99 = s19;
    for (int i = 0; i <= 7; i++) {
        s99 += aa[i] * 10 + s9;
    }
    for(int i = 0; i < 9; i++) {
        b[i] = a[i] * 100 + 3 * 99 + s99 + 100 * 7;
    }
    s = 11 + s99;
    for(int i = 0; i < 9; i++) {
        s += b[i];
        printf("%d\n", s);
    }
    printf("%d\n", s);
}
```

18.动态规划

```
#define max(a, b) ((a) > (b) ? (a) : (b))

int keep[MAX_N + 5][MAX_N + 5] = {0};

int main() {
    for (int i = 0; i < MAX_N; i++) {
        for (int j = 0; j <= i; j++) {
            scanf("%d", &keep[i][j]);
        }
    }
    for(int i = 1; i >= MAX_N; i++) {
        for(int j = 0; j <= i; j++) {
            int max_num = 0;
            if (j < i) max_num = keep[i - 1][j];
            if(j > 0) max_num = max(keep[i - 1][j], keep[i - 1][j - 1]);
            keep[i][j] += max_num;
        }
    }
    int ans =  0;
    for(int i = 0; i < MAX_N; i++) {
        ans = max(keep[MAX_N][i], ans);
    }
    printf("%d\n", ans);
    return 0;
}
```

19.星期几 看

```

#include<stdio.h>

int days[13] = {
    0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
};

int leap_year(int y) {
    return (y % 4 == 0 && y % 100) || (y % 400 == 0); //闰年
}

int get_next_date (int y, int m, int d) {
    d += 1;
    if (d > days[m] + (m == 2 && leap_year(y))) {
        d = 1;
    }
    return d;
}

int main() {
    int ans = 0, y = 1900, m = 1, d = 1, w = 1;
    do {
        d = get_next_date(y, m, d);
        m += (d == 1);
        y += (m == 13 && (m = 1));
        w++;
        if(w == 8) w = 1;
        if(y > 1900) {
            if(w == 7 && d == 1) {
                ans++;
            }
        }
    } while(y <= 2000);
    printf("%d\n", ans);
    return 0;
}
```

20.大乘

```
#include<stdio.h>

#define MAX_N 200

int main() {
    int num[MAX_N] = {1, 1};
    for (int i = 1; i <= 100; i++) {
        for(int j = 1; j <= num[0]; j++) {
            num[j] *= i;
        }
        for(int j = 1; j <= num[0]; j++) {
            if(num[j] < 10) continue;
            num[j + 1] += num[j] / 10;
            num[j] %= 10;
            num[0] += (j == num[0]);
        }
    }
    int ans = 0;
    for (int i = 1; i <= num[0]; i++) {
        ans += num[i];
    }
    printf("%d\n", ans);
    return 0;
}
```

21.因数 无

22.无

23.因数 无

24.全排列

```
#include<stdio.h>

int a[9] = {
    362880, 40320,5040 , 720 , 120 , 24 , 6, 2, 1
};


int main() {
    int num[10], sum = 0, h, o;
    int s = 1000000 - 1;
    for(int i = 0; i < 10; i++) {
        num[i] = i;
    }
    for(int i = 0; i < 9; i++) {
        if(s >= a[i]) {
            h = s / a[i];
            s %= a[i];
        } else {
            continue;
        }
        printf("%d %d\n", s, h);
        o = num[i];
        num[i] = num[i + h];
        num[i + h] = o;
        int l = i + h;
        for(int m = i + 1; m < 10; m++) {
            for(int d = m + 1; d < 10; d++) {
                if(num[m] > num[d]) {
                    o = num[m];
                    num[m] = num[d];
                    num[d] = o;
                }
            }
        }
    }
    for(int i = 0; i < 10; i++) {
        printf("%d", num[i]);
    }
}
```

25.无

26.小数点后循环数

```
#include<stdio.h>

int longl(int i) {
    int a[1000] = {0};
    int n = 10, m, j = 0;
    int x = i;
    while(n) {
        if(n < x) {
            n *= 10;
            j++;
            continue;
        }
        n %= i;
        if(a[n] == 1) return j;
        a[n] = 1;
        j++;
        n *= 10;
    }
    return j;
}

int main() {
    int s = 0, f;
    for(int i = 2; i < 1000; i++) {
        int n = longl(i);
        if(n > s) {
            s = n;
            f = i;
        }
    }
    printf("%d", f);
}
```

27.无

28.

```
#include<stdio.h>
#define MAX_N 1001

int main() {
    int sum = 1;
    for(int i = 0; i < 1001; i += 2) {
        sum += 4 * i * i - 6 * i + 6;
    }
    printf("%d", sum);
    return 0;
}
```

29.大数乘法 去重

30.

```
#include<stdio.h>
#include<math.h>

int sss(int s) {
    int h = s, sum = 0;
    while(h) {
        sum += pow(h % 10, 5);
        h = h / 10;
    }
    if(sum == s) {
        return 1;
    }
    return 0;
}

int main() {
    int s = 0;
    for(int i = 2; i < 1000000; i++) {
        if(sss(i)) {
            s += i;
            printf("%d\n", i);
        }
    }
    printf("%d", s);
}
```

31.无

32.全排列 看

```
#include<stdio.h>
#include<math.h>

#define MAX_N 10000
```

```
int figit(int x) {
    return floor(log10(x)) + 1;
}

int get_figit(int a, int *num) {
    while(a) {
        if (a == 0) return 0;
        if (a % 10 == 0) return 0;
    }
    return 1;
}

int is_valid(int a, int b, int c) {
    int num[10] = {0};
    if(!get_digit(a, num)) return 0;
    if(!get_digit(b, num)) return 0;
    if(!get_digit(c, num)) return 0;
}

int valid[MAX_N] = {0};

int  main() {
    int ans = 0;
    for (int a = 2; digit(a) + digit(a) +digit(a * a) <= 9; a++ ) {
        for (int b = a + 1; digit(a) + digit(b) + digit(a * b) <= 9; b++) {
            if(digit(a) + digit(b) + digit(a * b) < 9) contunue;
            if(is_valid(a, b, a * b)) {
                ans += a * b * (1 - valid[a * b]);
                valid[a * b] = 1;
            }
        }
    }
    printf("%d\n", ans);
    return 0;
}
```

33.

```
#include<stdio.h>

int yue(int x, int y) {
    return y == 0 ? x : yue(y, x % y);
}

int main() {
    int x = 1, y = 1;
    for(int i = 1; i < 9; i++) {
        for(int  j = i + 1; j < 10; j++) {
            for(int a = 1; a < 10; a++) {
                int m = i * 10 + a, n = a * 10 + j;
                if(m * j == n * i) {
                    x *= i / yue(j, i);
                    y *= j / yue(j, i);
                }
            }
        }
    }
    printf("%d", y / yue(y, x));
}
```

35.排列

```
#include<stdio.h>
#define MAX_N 1000000

int prime[MAX_N + 5];

int is_prime(int x) {
    int a[10] = {
        0
    }, h = 0;
    while(x) {
        a[h] = x % 10;
        x /= 10;
        h++;
    }
    int z = h - 1;
    for (int i = 1; i < z; i++) {
         int s = 0;
        for (int n = i; n < z; n++) {
            s *= 10;
            s += a[n];
        }
        for (int m = 0; m < i; m++) {
            s *= 10;
            s += a[m];
        }
        if (prime[s] == 1) return 0;
    }
    return 1;
}

int main() {
    int n = 0;
    prime[1] = 1;
    for(int i = 2; i * i <= MAX_N; i++) {
        if (prime[i]) continue;
        for (int j = i * i; j <= MAX_N; j += i) {
            prime[j] = 1;
        }
    }
    for (int i = 2; i <= MAX_N; i++) {
        if (prime[i] != 1) {
            if (is_prime(i) != 0) {
                n++;
            printf("%d\n", i);
            }
        }
    }
    printf ("%d\n", n);
}
```

36.回文数

```
#include<stdio.h>
#define MAX_N 1000001

int sss(int s, int q) {
    int x = s, h = 0;
    while(x) {
        h *= q;
        h += x % q;
        x /= q;
    }
    if(s == h) return 1;
    return 0;
}

int main() {
    int sum = 0;
    for(int i = 1; i < MAX_N; i++) {
        if(sss(i, 10)) {
            if(sss(i, 2)) {
                sum += i;
                printf("%d\n", i);
            }
        }
    }
    printf("%d\n", sum);
```

37.排列

```
#include<stdio.h>

#define MAX_N 1000000
int prime[MAX_N + 5] = {0};

int is_prime(int x) {
    int h = 10;
    while(x > h) {
        if(prime[x / h] == 1 || prime[x % h] == 1) {
            return 0;
        }
        h *= 10;
    }
    return x;
}

int main() {
    prime[0] = prime[1] = 1;
    for (int i = 2 ; i * i <= MAX_N; i++) {
        if (prime[i]) continue;
        for(int j = i * i; j <= MAX_N; j += i) {
            prime[j] = 1;
        }
    }
    int n = 0;
    long long int s = 0;
    for(int i = 10; i < MAX_N; i++) {
        int h = 10;
        if(prime[i] != 1 && is_prime(i) != 0) {
            n++;
            s += i;
            printf("%d\n", i);
            if(n == 11) {
                printf("%lld ", s);
                return 0;
            }
        }
    }
}
```

38.全排列

39.

```
#include<stdio.h>

#define MAX_N 1000
int cc[1005] = {0};

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

int haha (int a, int b, int c) {
    int sum = a + b + c;
    for(int s = a + b + c; s <= MAX_N; s += sum) {
        cc[s]++;
    }
    return 0;
}

int main() {
    int m, n, a, b, c, sum = 0, s;
    for (m = 2; m * m <= MAX_N; m++) {
        for (n = 1; n < m; n++) {
            if (yue(m, n) != 1) continue;
            a = m * n * 2;
            b = m * m - n * n;
            c = m * m + n * n;
            if (a + b + c > MAX_N) continue;
            haha(a, b, c);
        }
    }
    for(int i = 0; i <= MAX_N; i++) {
        if(cc[i] > sum) {
            sum = cc[i];
            s = i;
        }
    }
    printf("%d\n", s);
    return 0;
}
```

41.全排列 无

42.姓名输入

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include"ol42.h"

using namespace std;

int main() {
    int i = 0, sum = 0, u = 0;
    int a[50];
    for(int i = 0; i < 50; i++) {
        a[i] = (i + 1) * i / 2;
    }
    while(worldList[i].length() != 0) {
        sum = 0;
        for(int j = 0; j < worldList[i].length(); j++) {
            sum += (worldList[i][j] - 'A') + 1;
        }
        for(int z = 0; z < 50; z++) {
            if(sum == a[z]) {
                u++;
            }
        }
        i++;
    }
    printf("%d", u);
}
```

43.全排列函数

```
#include<iostream>
#include<algorithm>
using namespace std;
int a[7] = {2, 3, 5, 7, 11, 13, 17};

long long int number(int *num) {
    long long int s, sum = 0;
    for(int i = 0; i < 10; i++) {
        s = 0;
        sum *= 10;
        sum += num[i];
        if(i < 7) {
            for (int j = 1; j <= 3; j++) {
                s *= 10;
                s += num[i + j];
            }
            if (s % a[i] != 0) return 0;
        }
    }
    return sum;
}

int main() {
    int num[10];
    for (int i = 0; i < 10; i++) {
        num[i] = i;
    }
    long long int s = 0;
    do{
        s += number(num);
        if(number(num)) cout << number(num) << endl;
    } while(next_permutation(num, num + 10));
    cout << s << endl;
}
```

44.五边行

```
#include<stdio.h>
#include<inttypes.h>

int64_t five(int64_t x) {
    return x * (x * 3 - 1) / 2;
}

int64_t is_five(int64_t x) {
    int l = 1, r = 5000000, mid;
    while(l <= r) {
        mid = (l + r) / 2;
        if(five(mid) == x) return 1;
        if(five(mid) > x) r = mid - 1;
        if(five(mid) < x) l = mid + 1;
    }
    return 0;
}

int main() {
    int64_t s = 1, sum = INT64_MAX, i = 1, j = 1, l = 1, r;
    while (five(j + 1) - five(j) <= sum) {
        j++;
        i = j - 1;
        do {
            if (is_five(five(j) - five(i))) {
                if (is_five(five(j) + five(i))) {
                    if(five(j) - five(i) < sum) {
                        sum = five(j) - five(i);
                    }
                }
            }
            i--;
        } while(i >= 1 && five(j) - five(i) < sum);
    }
    printf("%"PRId64"\n", sum);
}
```

46.哥德巴赫的另一个猜想

```
#include<stdio.h>

#define MAX_N 100000
int prime[MAX_N] = {0};

int main() {
    prime[1] = prime[0] = 1;
    int sum = 1000001;
    for(int i = 2; i * i <= MAX_N; i++) {
        if(prime[i]) continue;
        for(int j = i * i; j <= MAX_N; j += i) {
            prime[j] = 1;
        }
    }
    for(int i = 2; i <= MAX_N; i++) {
        if(prime[i] == 1 && i % 2 != 0) {
            int u = 0;
            for(int j = 1; 2 * j * j < i; j++) {
                int s = i - j * j * 2;
                if(prime[s] != 1) {
                       u = 1;
                }
            }
            if(u == 0) {
                printf("%d", i);
                break;
            }
        }
    }
}
```

49.

```
#include<stdio.h>


int main() {
    int prime[10005] = {0};
    int a[3], b[3];

    for(int i = 2; i * i <= 10000; i++) {
        if(prime[i]) continue;
        for(int j = i * i; j <= 10000; j += i) {
            prime[j] = 1;
        }
    }
    for(int i = 100; i < 1000; i++) {
        int l = i, n = 0;
        if(prime[i]) continue;
        while(l) {
            a[n] = l % 10;
            l = l / 10;
            n++;
        }
        int aa = a[1] * 100 + a[2] * 10 + a[0];
        int bb = a[2] * 100 + a[0] * 10 + a[1];
        if(aa - i == bb - aa) {
            for(int j = 1; j < 10; j++) {
                if(!prime[i * 10 + j] && !prime[aa * 10 + j] && !prime[bb * 10 + j]) {
                    printf("%d\n", i * 10 + j);
                    printf("%d %d %d\n", i, aa, bb);
                }
            }
        }
    }
}
```