A. 细胞游戏

无算法，考查编程基本功

```cpp
#include <iostream>        // jk
#include <string.h>
#include <algorithm>

using namespace std;

char board[105][105];
int m, n, q;

int checkLogic(int i, int j) {
    int count = 0;
    int left = j - 1;
    int right = j + 1;
    int top = i - 1;
    int bottom = i + 1;
    for(int x = top; x <= bottom; x++){
        for(int y = left; y <= right; y++){
            count = board[x][y] == 1 || board[x][y] == -1 ? count + 1 : count;
        }
    }
    return board[i][j] == 1 ? (count == 3 || count == 4 ? 1 : -1) : (count == 3 ? -2 : 0);
}

void update() {
    for(int i = 1; i <= m; i++){
        for(int j = 1; j <= n; j++){
            board[i][j] = checkLogic(i, j);
        }
    }
    for(int i = 1; i <= m; i++){
        for(int j = 1; j <= n; j++){
            board[i][j] = board[i][j] == 1 || board[i][j] == -2 ? 1 : 0;
        }
    }
}

int main() {
    cin >> m >> n >> q;
    memset(board, 0, sizeof(board));
    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            cin >> board[i][j];
            board[i][j] -= '0';
        }
    }

    while (q--) update();

    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            cout << (int)board[i][j] << " \n"[j==n];
        }
    }
    return 0;
}
```

B. 嵌套深度

常规写法：

```cpp
#include<bits/stdc++.h>          // Enal

using namespace std;

int main()
{
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;++i){
        char c[200];
        scanf("%s",c);
        int cnt=0,maxnum=0;
        for(int j=0;j<strlen(c);++j){
            if(c[j]=='('){
                ++cnt;
            }else{
                --cnt;
            }
            if(cnt<0){
                puts("-1");
                break;
            }else{
                maxnum=max(cnt,maxnum);
            }
        }
        if(cnt==0){
            cout<<maxnum-1<<endl;
        }else if(cnt>0){
            puts("-1");
        }
    }
```

```
        return 0;
    }
```

用栈写提示：

(1) #号作为开始和结束的标志。

(2) 出现的凡是“左括号”，则进栈。

(3) 出现的是“右括号”

首先检查栈是否空？

  1. 若栈空，则表明该“右括号”多余

  2. 否则和栈顶元素比较？

    2.1 若相匹配，则栈顶“左括号出栈”

    2.2 否则表明不匹配

(4) 表达式检验结束时

  1. 若栈空，则表明表达式中匹配正确

  2. 否则表明“左括号”有余

C. 装水容器

n ≤100， 暴力O(n^2)：

```cpp
# include <iostream>    // kcxz

int a[105];

int main() {
    int T;
    std::cin >> T;

    while (T--) {
        int n;
        std::cin >> n;
        for (int i=0; i<n; i++) {
            std::cin >> a[i];
        }

        int t = 0;
        int max = 0;
        for (int i=0; i<n; i++) {
            for (int j=i+1; j<n; j++) {
                t = std::min(a[j], a[i]) * (j-i);
                if (t > max) {
                    max = t;
                }
            }
        }

        std::cout << max << std::endl;
    }

    return 0; }
```

O(n) 算法：

```cpp
# include <iostream>   // kcxz

int a[105];

int main() {
    int T;
    std::cin >> T;

    while (T--) {
        int n;
        std::cin >> n;
        for (int i=0; i<n; i++){
            std::cin >> a[i];
        }

        int t = 0;
        int max = 0;
        int left = 0;
        int right = n-1;
        for (int i=0; i<n; i++){
           if (left > right) {
               break;
           }

           t = std::min(a[left], a[right])*(right - left);
           if (t > max) {
               max = t;
           }

           if (a[left] < a[right]){
               left++;
           } else {
               right--;
            }
        }

        std::cout << max << std::endl;
    }

    return 0;
}
```

D. 大佬数

质因数，即既是因数又是质数。

算术基本定理：

任何一个大于 1 的自然数N, 如果N 不为质数，那么N 可以唯一分解成有限个质数的乘积。

```cpp
# include <iostream>

int main() {
    int n;
    std::cin >> n;

    int sum = 0;
    for (int i=2; i<=n; i++) {
        if (0 == n%i) {
            sum += i;

            while (0 == n%i) {
                n /= i;
            }
        }
    }

    if (sum > 610) {
        std::cout << "a sdl wsl";
    } else {
        std::cout << "tcl";
    }

    return 0;
}
```

暴力法：

```cpp
# include <iostream>    // kcxz

bool isPrime(int n) {
    if (n <= 1)
        return false;

    if (n <= 3)
        return true;

    if (n % 2 == 0 || n % 3 == 0)
        return false;

    for (int i = 5; i * i <= n; i = i + 6)
        if (n % i == 0 || n % (i + 2) == 0)
            return false;

    return true;
}

int main() {
    int n;
    std::cin >> n;

    int cnt = 0;
    for(int i=2; i<=n; i++) {
        if (n % i == 0) {
            if (isPrime(i)) {
                cnt += i;
            }
        }
    }

    if (cnt > 610) {
        std::cout << "a sdl wsl";
    } else {
        std::cout << "tcl";
    }

    return 0;
}
```

E. 能不能让我捧个杯啊

无算法，考查编程基本功

```
# include <iostream>   // kcxz ?
# include <cstdio>

bool vis[705];
int pos[705];

int main(){
   int n;
   int k;
    scanf("%d %d", &n, &k);

    int x;
    for (int i=1; i<=k; i++) {
       scanf("%d", &x);
        vis[x] = true;
   }

   int j = 0;
    for (int i=1; i<=n+k; i++) {
       if(!vis[i]) {
           pos[++j] = i;
       }
   }

    n /= 10;
    printf("%d %d\n%d %d\n%d %d\n", pos[1], pos[n], pos[n+1], pos[n*3], pos[n*3+1], pos[n*6]);

    return 0;
}
```

F. 多组输出斐波拉契

考查斐波拉契数列和模运算。

根据求模公式：

$$(a + b) \% c = (a\%c + b\%c) \% c$$

从而有：

$$F_n \% m = F_{n-1} \% m + F_{n-2} \% m = (F_{n-1} + F_{n-2}) \% m$$

$$(n \geq 3 \ \&\& \ n \in N)$$

只需要将每一项斐波拉契求模即可。

只有一秒的时间，需要将数据记忆下来，否则还是会超时。

另外，数据过大，请不要用 java 写，也不要用 C++ 中的 cin 和 cout

复杂度： O(n)

```cpp
# include <iostream>    // kcxz
# include <cstdio>

const int mod = 998244353;
const int maxn = 1e6+5;
int a[maxn];

void init() {
    a[0] = 0;
    a[1] = 1;
    a[2] = 1;
    for (int i=3; i<maxn; i++) {
        a[i] = (a[i-1] + a[i-2]) % mod;
    }
}

int main() {
    init();
    int T;
    scanf("%d", &T);

    while (T--) {
        int t;
        scanf("%d", &t);

        printf("%d\n", a[t]);
    }

    return 0;
}
```

矩阵快速幂： 请自行查阅相应知识
复杂度： O(nlogn)

```
// O(nlogn)
# include <iostream>

template <typename T>
void read(T &val) {
    T x = 0;
    int bz = 1;
    char c;

    for (c = getchar(); (c<'0' || c>'9') && c != '-';
    c = getchar());

    if (c == '-') {
        bz = -1;
        c = getchar();
    }

    for (; c >= '0' && c <= '9'; c = getchar())
        x = x * 10 + c - 48;

    val = x * bz;
}

template <typename T>
void put(T x){
    static char s[20];
    int bas;

    if(x < 0) {
        putchar('-');
```

```cpp
        x = -x;
    }

    if(x == 0) {
        putchar('0');
        return;
    }

    bas = 0;
    for(;x;x/=10)
        s[bas++] = x%10+'0';

    for(;bas--;)
        putchar(s[bas]);
}

const int mod = 998244353;
struct  Node {
    int a[2][2] = {{1,1},{1,0}};
    Node& operator* (Node& b ) {
        Node x;
        x.a[0][0] = 1ll * (1ll * this->a[0][0] * b.a[0][0] ) % mod + (1ll * this->a[0][1] * b.a[1][0] ) % mod;
        x.a[0][1] = 1ll * (1ll * this->a[0][0] * b.a[0][1] ) % mod + (1ll * this->a[0][1] * b.a[1][1] ) % mod;
        x.a[1][0] = 1ll * (1ll * this->a[1][0] * b.a[0][0] ) % mod + (1ll * this->a[1][1] * b.a[1][0] ) % mod;
        x.a[1][1] = 1ll * (1ll * this->a[1][0] * b.a[0][1] ) % mod + (1ll * this->a[1][1] * b.a[1][1] ) % mod;

        for (int i=0; i<2; i++) {
            for (int j=0; j<2; j++) {
                a[i][j] = x.a[i][j];
            }
        }

        return *this;
    }
};

Node qpow(Node x, int p) {
    Node ans;
    ans.a[0][0] = 1;
    ans.a[0][1] = 0;
    ans.a[1][0] = 0;
    ans.a[1][1] = 1;

    while (p) {
        if (p & 1) {
            ans = ans * x;
        }
        x = x * x;
        p >>= 1 ;
    }

    return ans;
}

void solve(int n) {
    if (n == 1 || n ==2){
        put(1);
        puts("");
        return;
    }

    Node a;
    a = qpow(a, n-1);
    put(a.a[0][0] % mod);
    puts("");
}

int main() {
    int T;
    read(T);

    while (T--) {
        int n;
        read(n);

        solve(n);
    }

    return 0;
}
```

G. 吃面包

考查 01 背包(0-1 Knapsack)方案数

```cpp
# include <iostream>   // NQ
# include <cstdio>
# include <algorithm>

using namespace std;

const int N = 1010, mod = 1e9 + 7;
int n, m, nums[N];

int main()
{
    scanf("%d %d", &n, &m);
    nums[0] = 1;
    while (n--) {
        int v;
        scanf("%d", &v);
        for (int i = m; i >= v; i--)
            nums[i] = (nums[i] + nums[i - v]) % mod;
    }
    cout << nums[m] << endl;
    return 0;
}
```

H. 能否购买

考查单调队列

```cpp
 # include <iostream>   // NQ
 # include <cstdio>
 # include <algorithm>

 using namespace std;

 const int N = 1e6 + 5;
 int n, m, k, sum[N];
 int q[N], hh, tt;

 int main()
 {
 scanf("%d  %d  %d",  &n,  &m,  &k);
 for  (int  i  =  1;  i  <=  n;  i++)  {
scanf("%d", &sum[i]);
sum[i] += sum[i - 1];
 }
int max_res = -0x3f3f3f3f;
for (int i = 1; i <= n; i++) {
        if (hh <= tt && i - q[hh] > k)
            hh++;
        max_res = max(max_res, sum[i] - sum[q[hh]]);
        while (hh <= tt && sum[i] <= sum[q[tt]])
            tt--;
        q[++tt] = i;
    }
    while (m--) {
        int x;
        scanf("%d", &x);
```

```c
        if (max_res >= x)
            printf("YES\n");
        else
            printf("NO\n");
    }
    return 0;
}
```