

# 递归

## 人员

柳力玮、韩昱辰、温郝冬、田心一、高健桓、纪博涵 到课, 初锦阳、蒋叔璋、刘宸熙 线上

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1211>

王向东老师周日三点半C++综合练习									
#	用户名	姓名	编程分	时间	A	B	C	D	
1	liulivei	柳力玮	400	10326	100	100	100	100	
2	gaojianhuan	高健桓	400	20042	100	100	100	100	
3	tianxinyi	田心一	400	22569	100	100	100	100	
4	jiangshuzhang	蒋叔璋	316	20601	100	100	100	16	
5	liruihan	李瑞涵	300	2036	100	100		100	
6	wenhaodong	温郝冬	300	21186	100	100	100		
7	hanyuchen	韩昱辰	206	18469	100	100	6		
8	chujin yang	初锦阳	205	17753	100	100	0	5	
9	lizhishuo	李知朔	200	10628	100	100	0		
10	liuchenxi	刘宸熙	120	14244	100	20			

## 本周作业

<https://cppoj.kids123code.com/contest/1343> (课上讲了 A ~ G 题, 课后作业是 F G 题)

## 课堂表现

今天给同学们讲了 递归 的内容, 主要练习了  $2^n$  枚举所有方案的题目

其中, 今天的 E F G 三道题目不要求同学们拿满分, 会用  $2^n$  枚举拿到 85 20 40 分即可

## 课堂内容

### New Place (上周作业)

求最长的后缀长度 len, 要求 s 中后 len 个字符的子串是 t 的子序列

求这个的过程可以从后往前扫, 用双指针实现

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

using namespace std;

int main()
{
```

```

int n; cin >> n;
string s, t; cin >> s >> t;

string s2 = s, t2 = t;
sort(s2.begin(), s2.end()), sort(t2.begin(), t2.end());
if (s2 != t2) { cout << -1 << endl; return 0; }

int cnt = 0;
for (int i = n-1, j = n-1; i >= 0; --i) {
    if (t[i] == s[j]) --j, ++cnt;
}
cout << n - cnt << endl;
return 0;
}

```

## 【深基7.例7】计算阶乘

递归求阶乘

```

#include <bits/stdc++.h>

using namespace std;

int f(int n) {
    if (n == 1) return 1;
    return n * f(n-1);
}

int main()
{
    int n; cin >> n;
    cout << f(n) << endl;
    return 0;
}

```

## 斐波那契数列

递归求斐波那契

```

#include <bits/stdc++.h>

using namespace std;

int f(int n) {
    if (n==1 || n==2) return 1;
    return f(n-1) + f(n-2);
}

int main()

```

```
{
    int T; cin >> T;
    while (T -- ) {
        int n; cin >> n;
        cout << f(n) << endl;
    }
    return 0;
}
```

## 数数小木块

递归求  $n$  层一共有多少小木块, 第  $i$  层有  $(1+i)*i / 2$  个小木块

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

using namespace std;

int f(int n) {
    if (n == 1) return 1;
    return (1+n)*n/2 + f(n-1);
}

int main()
{
    int n;
    cin >> n;
    cout << f(n) << endl;
    return 0;
}
```

## 枚举子集 (递归实现指类型枚举)

递归枚举第  $i$  个数选或者不选

```
#include<iostream>
using namespace std;
char s[105];
int x;
void dfs(int n){
    if(n==0){
        for(int i=x;i>=1;i--){
            cout<<s[i];
        }
        cout<<endl;
        return;
    }
    s[n]='N';
    dfs(n-1);
}
```

```

    s[n] = 'Y';
    dfs(n-1);
}
int main()
{
    cin >> x;
    dfs(x);
    return 0;
}

```

## [NOIP 2002 普及组] 选数

$2^n$  枚举每个数选或不选, 看有多少合法方案

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 20 + 5;
int w[maxn];
bool st[maxn];
int n, k, ans = 0;

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

void dfs(int u) {
    if (u == n+1) {
        int cnt = 0, sum = 0;
        for (int i = 1; i <= n; i++) {
            if (st[i]) {
                cnt++, sum += w[i];
            }
        }
        if (cnt == k && is_prime(sum)) ans++;
        return;
    }

    st[u] = false;
    dfs(u+1);
    st[u] = true;
    dfs(u+1);
}

int main()
{

```

```
cin >> n >> k;
for (int i = 1; i <= n; i++) cin >> w[i];
dfs(1);
cout << ans << endl;
return 0;
}
```