



程序设计实践（一）

哈尔滨工业大学 计算机学院
任课教师：孙大烈教授
助教：付万增



习题

- 一、基础
- 二、排序
- 三、贪心
- 四、分治



回文素数

打印 n ($2 \leq n \leq 1e9$) 以内的回文素数。

回文：正序和逆序相同，如**12321**和**1221**。

为了方便测试，只打印后**10**个。

对比程序：素数判断。

提示：直接枚举判断复杂度太高，但是回文数字并不多。

Input

第一行包含一个正整数 n ($2 < n \leq 1e5$)。

Output

n 以内的回文素数。



回文素数

```
void cal(int a,int b,bool odd)
{
    for(int i=1;;i++)
    {
        int top = 0,x = i,y = x;
        if(odd) x/=10;
        while(x)
        {
            y = y * 10 + x % 10;
            x /= 10;
        }
        if(y>b) break;
        if(y>=a && isPrime(y) ) ans[tot++] = y;
    }
}
```



筛法求素数

打印 n ($2 \leq n \leq 1e5$) 以内的素数。
为了方便测试，只打印后10个。
对比程序：素数判断，回文素数。

Input

第一行包含一个正整数 n ($2 < n \leq 1e5$)。

Output

n 以内的素数。



筛法求素数

```
const int N = 16000 + 10;
bool b[N];

void pre() //筛法求素数, b[i] = 0 表示 i 为素数
{
    for(int i=2;i<N;i++)
    {
        if(b[i] == 0)
        {
            for(int j=i+i;j<N;j+=i)
            {
                b[j] = 1;
            }
        }
    }
    b[1] = b[2] = 1;
}
```



筛法求素数应用

打印 n ($2 \leq n \leq 1e5$) 以内的每个数字的最大素因数。
为了方便测试，只打印后10个。
对比程序：筛法求素数。

Input

第一行包含一个正整数 n ($2 < n \leq 1e5$)。

Output

n 以内每个数字的最大素因数。



筛法求素数应用

```
for(int i=2;i<N;i++)
{
    if(b[i] == 0)
    {
        for(int j=i+i;j<N;j+=i)
        {
            b[j] = 1;
            ans[j] = i;
        }
        ans[i] = i;
    }
}
```




大大走格子

从 $n*m$ 的二维格子左上角走到右下角的方案数？

升级版问题：

二维格子中存在 k 个格子坏掉了，不能经过。

Input

第一行包含两个正整数 n, m ($2 < n, m \leq 1e3$)。

Output

方案数。



大大走格子

```
int n,m;  
scanf("%d%d",&n,&m);  
a[0][1] = 1;  
for(int i=1;i<=n;i++)  
{  
    for(int j=1;j<=m;j++)  
    {  
        a[i][j] = a[i-1][j] + a[i][j-1];  
    }  
}  
printf("%d\n",a[n][m]);  
return 0;
```



大大走格子加障碍

```
int x,y,n,m,k;
scanf("%d%d",&n,&m);
a[0][1] = 1;
scanf("%d",&k);
while(k--)
{
    scanf("%d%d",&x,&y);
    f[x][y] = 1;
}
for(int i=1;i<=n;i++)
{
    for(int j=1;j<=m;j++)
    {
        if(f[i][j]) continue;
        a[i][j] = a[i-1][j] + a[i][j-1];
    }
},
```



背包问题增强版

小李有一个可以容纳 V kg的背包。现在有 n 种砂石，每种总含量为 a_i kg，性价比 b_i bit/kg。问小李的背包最多容纳多少bit的物品？

样例输入：(按照 V N a_i b_i 的顺序输入)

100 3

60 8

30 10

80 3

样例输出：(30*10+60*8+10*3=)810



背包问题增强版

```
maopao(a,b,n);  
//for(int i=0;i<n;i++) printf("%d%",  
int ans = 0;  
for(int i=0;i<n;i++)  
{  
    if(V>=a[i])  
    {  
        ans += a[i] * b[i];  
        V -= a[i];  
    }else  
    {  
        ans += V * b[i];  
        break;  
    }  
}
```



变大的数字

给出一个十进制数字，可以将该数字相邻的两个数字进行一次交换，问交换以后可能变成的最大数字是多少？

Input

第一行包含一个正整数 n ($0 < n \leq 10000$)。

Output

一行一个整数表示交换以后可能生成的最大整数。



变大的数字

```
int n, top = 0;
scanf("%d", &n);
while(n > 0) //利用数组存储n十进制的每一位
{
    a[top] = n % 10;
    n /= 10;
    top++;
}
for(int i=top-1; i>0; i--)
{
    if(a[i] < a[i-1])
    {
        int t = a[i];
        a[i] = a[i-1];
        a[i-1] = t;
        break;
    }
}
```



吃糖果

聪聪有 n 种糖果，每种糖果 a_i 颗。聪聪吃糖果时有个特殊的癖好，就是不喜欢将一样的糖果放在一起吃，喜欢先吃一种，下一次吃另一种，这样；可是聪聪不知道是否存在一种吃糖果的顺序使得他能把所有糖果都吃完？

Input

第一行包含一个正整数 n ($2 < n \leq 1e4$)。

第二行包含 n 个正整数 a_i ($2 < a_i \leq 1e4$)。

Output

Yes or No.



吃糖果

```
int n,T,x;
scanf("%d",&T);
while(T--)
{
    scanf("%d",&n);
    long long sum = 0,mx = 0;
    while(n--)
    {
        scanf("%d",&x);
        if(x>mx) mx = x;
        sum += x;
    }
    printf("%s\n",sum-mx+1LL>=mx?"Yes":"No");
}
```



友谊的小船

n 个人，已知每个人体重。独木舟承重固定，每只独木舟最多坐两个人，可以坐一个人或者两个人。显然要求总重量不超过独木舟承重，假设每个人体重也不超过独木舟承重，问最少需要几只独木舟？

Input

第一行包含两个正整数 n ($0 < n \leq 100$)和 m ($0 < m \leq 2000$)，表示人数和独木舟的承重。

接下来 n 行，每行一个正整数，表示每个人的体重。体重不超过1000，并且每个人的体重不超过 m 。

Output

一行一个整数表示最少需要的独木舟数。



友谊的小船

```
maopao(a,n);  
//贪心原则：优先放置体重比较大的人，仔细想想为什么？  
  
for(int i=0;i<n;i++) printf("%d%c",a[i],i==n-1?'\\n':' ');  
  
int ans = 0;  
for(int i=0;i<n;i++)  
{  
    if(!b[i]) continue; //已经放置过的人直接continue  
    b[i] = 0; //该同学一定放置  
    ans++;  
    //printf("%d %d\\n",i,a[i]);  
    for(int j=i+1;j<n;j++)  
    {  
        if(!b[j]) continue;  
        if(a[i] + a[j] <= m)  
        {  
            //找到刚好a[i] + a[j] <= m的i和j，为i和j分配独木舟，这样利用率最高，想想为什么  
            b[j] = 0;  
            break;  
        }  
    }  
}
```



完美字符串

约翰认为字符串的完美度等于它里面所有字母的完美度之和。每个字母的完美度可以由你来分配，不同字母的完美度不同，分别对应一个1-26之间的整数。约翰不在乎字母大小写。（也就是说字母F和f）的完美度相同。给定一个字符串，输出它的最大可能的完美度。例如：**dad**，你可以将**26**分配给**d**，**25**分配给**a**，这样整个字符串完美度为**77**。

字符串是什么？ 字符的数组。

定义： `char s[100];`

输入： `scanf("%s",s);`

赋值： `s[0] = 'a';`

输出： `printf("%s",s);`



完美字符串

```
scanf("%s",s);
int len = strlen(s); //求s的长度
for(int i=0;i<M;i++) num[i] = 0;
for(int i=0;i<len;i++)
{
    if(s[i]>='A' && s[i]<='Z')
    {
        num[s[i] - 'A']++;
    }else
    {
        num[s[i] - 'a']++;
    }
}
maopao(num,M);
int ans = 0;
for(int i=0;i<M;i++)
{
    if(num[i]) printf("%c出现了%d次\n",i+'a',num[i]);
    ans += num[i] * (M-i);
}
printf("字符串%s的完美度为: %d\n",s,ans);
```

小结



重点掌握

- 1.基础：筛法求素数
- 2.递推：大大走格子
- 3.排序：背包问题
- 4.贪心：独木舟
- 5.基数排序：完美字符串

什么是递归？

- 想象你拿着一面镜子站在另一面镜子面前
- 想象你在大峡谷里喊一声“Hello, World!”
- 程序调用自身的编程技巧称为递归（recursion）。
- 一般来说，递归需要有边界条件。

吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我



递归



```
#include <stdio.h>
void Recursion(int depth){
    printf("抱着");
    if (!depth) printf("我的小鲤鱼");
    else Recursion(--depth);
    printf("的我");
}
int main(){
    printf("吓得我抱起了\n");
    Recursion(2);
    putchar(' \n' );
}
```

吓得我抱起了

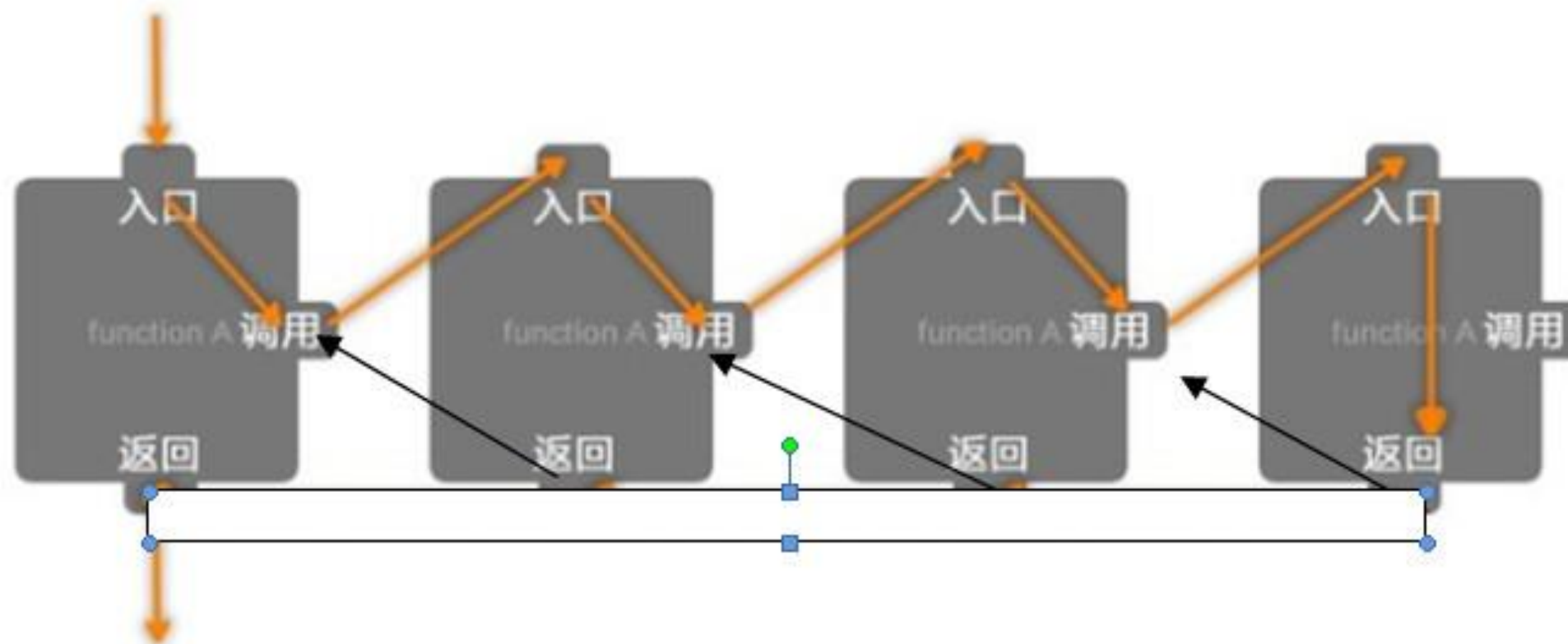
抱着抱着抱着我的小鲤鱼的我的我的我



递归



递归过程中发生了什么？





对比递归和递推（分别写出两种求斐波那契数列的程序）

- `int func(int x) //返回值类型声明 子程序名（参数）`
- `{`
 - `if(x == 0) return 1;`
 - `If(x == 1) return 1;`
 - `return func(x-1) + func(x-2);`
- `}`
-
- `for(int i=2;i<=n;i++)`
- `f[i] = f[i-1] + f[i-2];`

递推改递归

问题1: 求解斐波那契额数列 $f[n] = f[n-1] + f[n-2]$

问题2: 求解大大走格子 $f[n][m] = f[n-1][m] + f[n][m-1]$

问题3: 递归打印“我的小鲤鱼”

要求输入 n , 打印 n 个“抱着” + “我的小鲤鱼” + n 个“的我”

吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我





快速幂

求解 $x^n \bmod m = ?$ ($1 \leq x, n, m \leq 1e9$)

递推？更快的办法？

提示：计算 x, x^2, x^4, x^8 分别需要计算多少次乘法？

那么，计算 x^{10} 呢？

$10 = 1010(2)$

$x^{10} = x^8 * x^2$

Input

第一行包含三个正整数 x, n, m ($1 \leq x, n, m \leq 1e9$)。

Output

$x^n \bmod m = \text{ans}$



快速幂

```
int pow(int x,int n,int m)
{
    if(n==0) return 1;
    if(n==1) return x % m;
    int y = pow(x,n/2,m);
    if(n%2==0) return (y*y)%m;
    else return (y*y*x)%m;
}
```



1.hoj_A: 连续上升子序列长度

2.hoj_B: 最近的距离

3.hoj_C: 2倍增长与3倍增长

4.hoj_D: 回文素数

5.hoj_E: 水果 <http://acm.hit.edu.cn/hojx/showcontest/9/E/>

6.hoj_F: 直线分割平面



重点掌握（理解代码，模仿重复）

- 1.筛法求素数
- 2.完美字符串
- 3.快速排序
- 4.归并排序
- 5.快速幂



- 一分耕耘，一分收获！
- 希望在今后的学习生活中共同进步！

谢谢！