

题解

题解

重点

易错

写在前面

输入与输出

T1 Hello world

T3简单输入与输出

数据类型与算术运算符

算数运算符

数据类型

T4 A+B

T5求x的立方

T6输出字符的ASCII

T7计算圆柱侧面积和体积

条件与循环语句

关系运算符

T8整数比大小

T9大小写字母转换

T10一年级的算术题

T11判断三角形形状

T12时钟夹角

T14输出一行星号

数组

T13根据日期求星期

T15求解一元二次方程

T16 求最大素数

T17再做一年级算术题

T18 确定母亲节

T19 再算日期

函数与递归

T20 贪吃的猴子

T21三做一年级算术题

T22求最小的m值

图形题

T23数字正方形

T24数字菱形

T25倒三角形

T26计算sum的值

T27 寻找特殊偶数
T28.寻找阿姆斯特朗数
T29 零钱换整钱*
T30计算天数
T31 计算最后三位数
T32 孪生素数
T33计算后续日期
T34哥俩好
T35北理工的恶龙
T36谁能出线
T37大家一起做游戏
T38等值数列段
T39
复杂度
随着问题规模n的不断增大，上述时间复杂度不断增大，算法的执行效率越低
T40视频
T41音频
T42 汉字储存
T43 n位正整数的个数
T44 合并字符串
T45 删除重复字符
T46 组成最大数
T47
T48
T49再算天数
T50 求最大公约数
T51 求序列和
T52 函数的递归
T53回文字符串
T54 子串反转
T55多少种表示方法
T56 单词排序
T57
T58
T59
T60
T61
T62
T63
T64
T65
T66

重点

写代码注意缩进 换行 与大括号

易错

1. 分号
2. 中英文
3. 输入时的 `&` , `""`
4. 一个等号是赋值, 两个等号是判断相等
5. 数组下标从0开始
6. 数组不要开太小

写在前面

1. 函数的声明一定要在主函数之外
2. 例如:
3. `int max(int n,int m)`
4. `{`
- 5.
6. `}`
7. `int main()`
8. `{`
- 9.
10. `}`
- 11.
- 12.
13. 不支持以下方式
14. `int main()`
15. `{`
16. `int max(int n,int m){}`
17. `/*`
18. 有的语言支持内部类 到是可以这么写
19. `*/`
20. `}`
- 21.
22. `=` 表示赋值 // 至少没有重载的话 表示赋值
23. `==` 表示相等 // 至少没有重载的话 表示相等
- 24.
25. `if for while dowhile` 等 这些的内部的代码语句 如果只有一句 可以不加大括号
26. 如果有多句 一定要加, 而且注意缩进
27. 比如
28. `for(int i=1;i<=n;i++)`
29. `{`
30. `for(int j=1;j<=n;j++)`
31. `{`
32. `for(int k=1;k<=n;k++)`
33. `{`
34. `if(i!=j&&j!=k)`
35. `{`
36. `f[i][j]=min(f[i][j],f[i][k]+f[k][j]);`
37. `}`

```
38.     }
39. }
40. }
41.
42.
43. strlen函数 可以得到数组的使用了的长度
44. 比如
45. T21 中的
46. strlen(s+1)得到s数组用了多少
47. 因为我s数组下标从1开始用的 所以strlen(s+1) 里面有个加一
48. 如果是从0开始用的,就strlen(s)
49.
50.
51.
52. 读入字符数组 可以用%s
53. 例如
54. scanf("%s",s); 切记这里没有&
```

输入与输出

T1 Hello world

```
1. #include <stdio.h> //头文件
2. int main() //主函数
3. {
4.     printf("Hello world.\n"); //输出 在printf语句中 " "中的内容 除了有特别
    含义的 其余都是按原文输出
5.     /*
6.     特殊含义的: \n \r
7.     %d %c %x 这一种叫做占位符,要和输出的变量数据类型对应
8.     %d为整数 %02d表示输出两位不足两位用0补
9.     %lf 为浮点数 %.2lf为保留两位小数
10.    */
11.    return 0;
12. }
```

T3简单输入与输出

```
1. #include<stdio.h>
2. int main()
3. {
4.     int x;
```

```
5.     scanf("%d",&x); //在绝大部分情况读入数据时，都要加&（取址符） 另外一小部分
    是读入字符串
6.     //",&x这里的逗号一定要写
7.     printf("x=%d,x=%o,x=%x\n",x,x,x);
8. }
```

数据类型与算术运算符

算数运算符

+ - * / %
+= -= *= /= %=

++ --
ps:能用乘法不要用除法，能不取模就不取模 因为除法和取模会比较耗费时间
下面是运算符的优先级，请手动忽略表格第二行

↑表格按运算符优先级从高到低列出各个运算符，具有较高优先级的运算符出现在表格的上面，具有较低优先级的运算符出现在表格的下面。在表达式中，较高优先级的运算符会优先被计算。

类别	运算符	结合性
后缀	() [] -> . ++ --	从左到右
一元	+ - ! ~ ++ -- (type)* & sizeof	从右到左
乘除	* / %	从左到右
加减	+ -	从左到右
移位	<< >>	从左到右
关系	< <= > >=	从左到右
相等	== !=	从左到右
位与 AND	&	从左到右
位异或 XOR	^	从左到右
位或 OR		从左到右
逻辑与 AND	&&	从左到右
逻辑或 OR		从左到右
条件	?:	从右到左
赋值	= += -= *= /= %= >>= <<= &= ^= =	从右到左
逗号	,	从左到右

数据类型

整数类型

下表列出了关于标准整数类型的存储大小和值范围的细节：

类型	存储大小	值范围
char	1 字节	-128 到 127 或 0 到 255
unsigned char	1 字节	0 到 255
signed char	1 字节	-128 到 127
int	2 或 4 字节	-32,768 到 32,767 或 -2,147,483,648 到 2,147,483,647
unsigned int	2 或 4 字节	0 到 65,535 或 0 到 4,294,967,295
short	2 字节	-32,768 到 32,767
unsigned short	2 字节	0 到 65,535
long	4 字节	-2,147,483,648 到 2,147,483,647
unsigned long	4 字节	0 到 4,294,967,295

浮点类型

下表列出了关于标准浮点类型的存储大小、值范围和精度的细节：

类型	存储大小	值范围	精度
float	4 字节	1.2E-38 到 3.4E+38	6 位小数
double	8 字节	2.3E-308 到 1.7E+308	15 位小数
long double	16 字节	3.4E-4932 到 1.1E+4932	19 位小数

ps:int有2字节或者4字节是由计算机的32位还是64位决定的
乐学的题目到30多还没有爆int的
尽量不要用单精度
就算是双精度也是会有误差的，所以能用整数的地方不要用浮点数

T4 A+B

```
1. #include<stdio.h>
2. int main()
3. {
4.     int x,y;
5.     scanf("%d%d",&x,&y);
6.     printf("%d+%d=%d\n",x,y,x+y);
7. }
```

```
1.     a++; 先使用a的值，再执行a=a+1
2.     b=a++;等价于 b=a,a=a+1;
3.     ++a; 先执行a=a+1，再使用a的值
4.     b=++a;等价于 a=a+1,b=a;
```

T5求x的立方

```
1. #include<stdio.h>
2. int main()
3. {
4.     int x;
5.     scanf("%d",&x);
6.     printf("%d\n",x*x*x); //c不支持^表示多少次方，c的^表示异或
7. }
```

T6输出字符的ASCII

```
1. #include<stdio.h>
2. int main()
3. {
4.     char c;
5.     scanf("%c",&c);
6.     printf("The ASCII of character '%c' is %d.\n",c,c);
7.     //printf语句中的%d 会强制类型转换把字符c转换成对应的ascii值然后输出
8.     //ascii为字符与数字的一对一的映射，前256个为数字 字母 符号，后面还有汉字的
9.     //字符的加减法为对应ascii的加减法
10. }
```

T7计算圆柱侧面积和体积

```
1. #include<stdio.h>
2. double r,h;
3. int main()
4. {
5.     scanf("%lf%lf",&r,&h);
6.     printf("s=%.2lf,v=%.2lf\n",2*3.1415926*r*h,r*r*3.1415926*h);
7. }
```

条件与循环语句

关系运算符

<
>

<=

>=

== (一个等号是赋值，两个等号是判断相等)

!=

```
1.      if(表达式1)
2.      {
3.          执行操作语句      //如果表达式1为真，则进入if语块，执行操作语句
4.          //如果操作语句只有一句话（只有一个分号）则可以不加大括号
5.      }
6.
7.
8.      if (表达式1)
9.      {
10.         执行操作语句1
11.     }
12.     else
13.     {
14.         执行操作语句2
15.     }
16.     如果表达式1为真就执行操作语句1 否则执行操作语句2
17.     if (表达式1)
18.     {
19.         执行操作语句1
20.     }
21.     else if (表达式2)
22.     {
23.         执行操作语句2
24.     }
25.     else if (表达式3)
26.     {
27.         执行操作语句3
28.     }
29.     else
30.     {
31.         执行操作语句4
32.     }
33.     如果表达式1为真就执行操作语句1 否则看表达式2是否为真
34.     如果表达式2为真则执行操作语句2，否则看表达式3是否为真
35.     如果表达式3为真则执行操作语句3，否则执行else的语句
36.     else if可以写任意多个
37.
38.
39.
40.
41.     三目运算符
42.     A?B:C;
43.     若A为真就返回B否则返回C
44.
45.     switch 语句 这个我很少用到
```



```
46.      switch(x)根据x值的不同执行不同的语句
47.      #include <stdio.h>
48.
49.  int main ()
50.  {
51.      char grade = 'B';
52.
53.      switch(grade)
54.      {
55.          case 'A' :
56.              printf("很棒! \n");
57.              break;
58.          case 'B' :
59.              printf("好! \n");
60.              break;
61.          case 'C' :
62.              printf("做得好\n" );
63.              break;
64.          case 'D' :
65.              printf("您通过了\n" );
66.              break;
67.          case 'F' :
68.              printf("最好再试一下\n" );
69.              break;
70.          default :
71.              printf("无效的成绩\n" ); //类似于if语句的else
72.      }
73.      return 0;
74.  }
75.
76.
77.
78.  for(x1;x2;x3)
79.  {
80.      x4;
81.  }
82.  1.执行x1
83.  2.判断x2是否为真, 若为真进入第三步, 否则结束循环
84.  3.执行x4
85.  4.执行x3, 返回第二步
86.  例子:
87.  for(int i=1;i<=n;i++)
88.      sum+=i;
89.  //对1到n求和
90.  while(x1)
91.  {
92.      x2
93.  }
94.  1.判断x1是否为真, 若为真进入第二步, 否则结束循环
95.  2.执行x2 返回第一步
96.  while(i<=n)
97.  {
```

```

98.     sum+=i;
99.     i++;
100. }
101. do
102. {
103.     x2
104. }while(x1)
105. 1.执行x2
106. 2.判断x1是否为真，若为真进入第一步，否则结束循环
107. do
108. {
109.     sum+=i;
110. }while(i<=n)
111.
112. continue;表示跳过当前循环执行下一次的循环
113. break;表示结束本层的循环
114.
115. for(int i=1;i<=5;i++)
116. {
117.     if(i==3)continue;//此时执行上面所写for语句执行过程的第四步
118.     if(i==4)break;//此时直接结束本层循环
119. }

```

T8整数比大小

```

1. #include<stdio.h>
2. int n,m;
3. int main()
4. {
5.     scanf("%d%d",&n,&m);
6.     printf("min=%d\n",n>m?m:n); //如果n大于m则返回m 否则返回n
7. }

```

T9大小写字母转换

```

1. #include<stdio.h>
2. char x;
3. int main()
4. {
5.     scanf("%c",&x);
6.     if(x>='a'&&x<='z')printf("%c\n",x-32); //如果x是小写字母
7.     else if(x>='A'&&x<='Z')printf("%c\n",x+32); //如果x是大写字母
8.     else printf("%c\n",x); //如果x是非字母的字符
9.     //字符类型比较大小 转换成ascii比较
10.    //加减32是因为同一字母大小写的ascii值差32

```

```
11.         //例如a的值是97 A的值是65
12.     }
```

T10一年级的算术题

```
1. #include<stdio.h>
2. int n,m;
3. int main()
4. {
5.     scanf("%d%d",&n,&m);
6.     printf("%d+%d=%d\n%d-
%d=%d\n%d*%d=%d\n%d/%d=%d\n%d%%d=%d\n",n,m,n+m,n,m,n-
m,n,m,n*m,n,m,n/m,n,m,n%m);
7.     //printf语句中模运算是写的 %d% % %d=%d
8.     //是因为%符号是有特殊含义的，所以需要转义，所以用第二个%把第三个%的特殊含义去除
掉
9.     //类似的转义字符还有\
10. }
```

T11判断三角形形状

```
1. #include<stdio.h>
2. int a,b,c;
3. int max(int n,int m){return n>m?n:m;} //自定义函数，功能是返回两个值的最大值
4. int min(int n,int m){return n>m?m:n;} //自定义函数，功能是返回两个值的最小值
5. int main()
6. {
7.     scanf("%d%d%d",&a,&b,&c);
8.     int mx=max(a,max(b,c)),mn=min(a,min(b,c)); //mx是得到三角形的最长边，m
n得到三角形的最短边
9.     int mm=a+b+c-mx-mn; //mm是得到三角形第二长边
10.    if(mn+mm<=mx)printf("non-triangle.\n"); //两短边之和小于等于最长边
11.    else if(mx==mn)printf("equilateral triangle.\n"); //最长边等于最短边
12.    else if(mm==mn||mx==mm)printf("isoceses triangle.\n"); //次长边等于
最短边或者最长边等于次长边
13.    else printf("triangle.\n");
14. }
```

T12时钟夹角

```
1. #include<stdio.h>
2. int n,m;
```

```

3. double cal(double res){if(res<0)res=-res;return res;} //自定义函数，得到一个数的绝对值
4. int main()
5. {
6.     scanf("%d%d",&n,&m);
7.     printf("At %d:%02d the angle is %.1lf degrees.\n",n,m,cal(6.00*m-((n%12)*30.00+30*(m/60.00))));
8.
9.     // 6.00*m得到分针的度数（规定指向12时为0度）
10.    // (n%12)*30.00+30*(m/60.00)为计算时针的度数（规定同上）
11.    // 两个度数相减便是夹角度数
12.    //因为度数为正，所以要对减法得到的结果进行处理
13.    //如果结果小于0，就取相反数输出
14. }
15. //以及这题乐学数据比较水，没有出现减法得到结果是超过180度的，（也有可能是我智障 没有想到不会出现大于180度的情况）所以代码中没有判断超过180度该怎么办

```

T14输出一行星号

```

1. #include<stdio.h>
2. int n,m;
3. int main()
4. {
5.     scanf("%d",&n);
6.     for(int i=1;i<=n;i++)printf("*");printf("\n");
7. }

```

数组

定义:表示相同数据类型的集合

```

1. 声明方法
2. int a[1000];
3. //定义了一个叫做a的数组，大小为1000，数据类型为int
4. //大小为1000，说明0到999可用，1000是不可用
5. //a[-1],a[1000]都会数组越界
6. int a[]={1,2,3,4,5,6,7,8,9};
7. //这样声明也是合法的，切记{}中第一个元素放在了a[0]中 第二个元素放在a[1]中
8.
9. 数组数据的读入
10. int a[1000];
11. for(int i=1;i<=n;i++)scanf("%d",&a[i]);
12.
13.
14. 下面讲字符数组的读入

```

```

15. char s[10000];
16. scanf("%s",s); //这样读入会读入一整行字符，， 切记这里没有&
17. //而且这样读入第一个元素同样会放在s[0]
18. scanf("%s",s+1) //这样读入， 第一个元素会放在s[1]

```

T13根据日期求星期

```

1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},n,m,k,sum;
3. int main()
4. {
5.     //a数组用来放每个月的天数
6.     scanf("%d%d%d",&n,&m,&k);
7.     for(int i=1900;i<n;i++)if((i%4==0)&&(i%100!=0)||
(i%400==0))sum+=366;else sum+=365; //从1900循环到n-1年， 如果为闰年则天数加3
66， 如果为平年天数加365
8.     if((n%4==0)&&(n%100!=0)|| (n%400==0))a[2]=29; //如果第n年为闰年， 第二个
月改成29天
9.     for(int i=1;i<=12;i++)a[i]+=a[i-1];
10.    sum+=a[m-1];
11.    /*
12.    for(int i=1;i<m;i++)sum+=a[i]; 这样写和9.10行写法功能是相同的， 都是求1到m
-1月有多少天
13.    */
14.    sum+=k; //算当前月过了多少天
15.    printf("%d\n",(sum)%7);
16. }

```

写到这发现我自己不会分类子

T15求解一元二次方程

```

1. #include<stdio.h>
2. #include<math.h>
3. #include<complex.h>
4. int a,b,c;
5. int main()
6. {
7.     scanf("%d%d%d",&a,&b,&c);
8.     double x=b*b-4*a*c;
9.     if(a==0&&b==0)printf("Input error!\n");
10.    else if(x<0) { //求虚数解
11.        if(b!=0)printf("x1=%.6lf+%.6lfi\x2=%.6lf-%.6lfi\n",(-b)/(2.00*
a),(sqrt(-x)/(2.00*a)),(-b)/(2.00*a),(sqrt(-x)/(2.00*a)));

```

```

12.         else printf("x1=%.6lf\nx2=-%.6lf\n", (sqrt(-x)/(2.00*a)),
    (sqrt(-x)/(2.00*a)));}
13.         else if(a==0)printf("x=%.6lf\n", (-c*1.00)/(b*1.00)); //一次式
14.         else if(x==0)printf("x1=x2=%.6lf\n", (-b)/(2.00*a)); //二次式 一解
15.         else printf("x1=%.6lf\nx2=%.6lf\n", (-b+sqrt(x))/(2.00*a), (-b-
    sqrt(x))/(2.00*a)); //二次式两解
16.
17. }

```

T16 求最大素数

```

1. #include<stdio.h>
2. int can(int x)
3. {
4.     for(int i=2;i*i<=x;i++) //枚举因子
5.         if(x%i==0) //如果能整除
6.             return 0; //则不是素数 返回0
7.     return 1; //是素数返回1
8. } //can函数用来判断一个数是否是素数
9. int main()
10. {
11.     int n;
12.     scanf("%d",&n);
13.     while(n)
14.     {
15.         if(can(n)){printf("The max prime number is %d.\n",n);return 0;}
16.         n--;
17.     }
18.
19. }

```

T17再做一年级算术题

然鹅我并没有用switch

```

1. #include<stdio.h>
2. #include<math.h>
3. char s[100],x;
4. int main()
5. {
6.     scanf("%s",s+1);
7.     // 这里并没有 写& 我也忘记上面有没有解释为啥这时候不加&了
8.     //那就在这里解释
9.     // 加&是要得到你想用的变量的地址，然后才能存数据

```

```

10. // 数组名就相当于地址，所以这里就并没有加
11. // 重点：如果是a[i] 格式则必须加& 如果是a+i则不加&
12. /*
13. scanf("%d",a+1); scanf("%d",&a[i]); 这两句等价
14. printf("%d\n",a[1]); printf("%d\n",*(a+1)); 这两句等价
15. printf("%d\n",a+1); 这样输出是得到a[1]的地址不是a[1]的值
16. */
17. int L=1,res=0,res2=0;
18. while(s[L])
19.     if(s[L]>='0'&&s[L]<='9')
20.         res=res*10+s[L++]-'0';
21.         else break;
22. x=s[L],L++;
23. while(s[L])res2=res2*10+s[L++]-'0';
24.
25. //我的读入方式写的有点另类
26. /*
27.     int x,y;
28.     char opt;
29.     scanf("%d%c%d",&x,&opt,&y);
30.     这样写是更优美简洁的
31. */
32. if(x=='+')printf("%d\n",res+res2);
33. else if(x=='-')printf("%d\n",res-res2);
34. else if(x=='*')printf("%d\n",res*res2);
35. else if(x=='/')printf("%d\n",res/res2);
36. else printf("%d\n",res%res2);
37. }

```

T18 确定母亲节

```

1. #include<stdio.h>
2. int can(int n)
3. {
4.     if((n%4==0&& n%100!=0) || n%400==0) return 1;
5.     return 0;
6. }
7. int main()
8. {
9.     int year,num=1,i;
10.    scanf("%d",&year);
11.    for(i=1601;i<=year;i++)
12.    {
13.        num+=365;
14.        if(can(i))num++;
15.        num%=7;
16.    }
17.    int tot=0,day=1;

```

```

18.     if(num==0)tot++; //tot表示周天的次数
19.     while(tot!=2)
20.     {
21.         num++,day++;
22.         if(num%7==0)tot++;
23.     }printf("%d\n",day); //当周天的次数为2时 结束循环，这一天便是母亲节
24.     return 0;
25. }

```

T19 再算日期

```

1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},n,m,k,sum;
3. int can(int i){if((i%4==0)&&(i%100!=0)|| (i%400==0))return 1;return 0;}

4. int main()
5. {
6.     scanf("%d%d%d",&n,&m,&k);
7.     for(int i=1900;i<n;i++)if(can(i))sum+=366;else sum+=365;
8.     if(can(n))a[2]=29;
9.     if(m<1||m>12){printf("month is error.\n");return 0;}
10.    if(k<1||k>a[m]){printf("day is error.\n");return 0;}
11.    for(int i=1;i<=12;i++)a[i]+=a[i-1];
12.    sum+=a[m-1];sum+=k;
13.    printf("%d\n",(sum)%7);
14.    //相比上一次算日期 就加了判断输入是否合法
15. }

```

函数与递归

```

1. 自定义函数格式
2. 函数数据类型 函数名（[参数]）
3. {
4.     return 返回值
5. }
6. 函数名命名规则和变量命名规则一致
7. []内容表示可有可无，根据自己需求来定
8. 如果有参数 则写参数的数据类型与参数名，多个参数的话要用逗号隔开
9. ex: int max(int n,int m,int x,int y)
10. 函数数据类型要和返回值的类型相同
11.
12. 参数只是局部变量，只会在函数这一部分有定义，不会影响调用函数时，传参的原变量，感觉有点绕口
13. ex:
14. void plus(int x)

```



```

15. {
16.     x++;
17. }
18. int main()
19. {
20.     int x=10;
21.     plus(x);
22.     printf("%d\n",x); //这里输出的x 还会是10，不是11，因为我们传参数，是把x的值
    传进去。
23. }
24.
25. 递归
26. 即函数自己调用自己
27.
28. int find(int x)
29. {
30.     if(x==1) return 1;
31.     return x+find(x-1);
32. }
33.
34. 类似于数列的递推式

```

$$a_n = n + a_{n-1}$$

$$a_1 = 1$$

$$a_n = n + a_{n-1} = n + n - 1 + a_{n-2} =$$

$$n + n - 1 + n - 2 + a_{n-3} = \dots = n + n - 1 + n - 2 \dots + 1$$

T20 贪吃的猴子

```

1. #include<stdio.h>
2. int n;
3. int find(int x)
4. {
5.     if(x==n) return 1;
6.     else return (find(x+1)+x)*2;
7. }
8. /*
9.     int a[1000];
10.    a[n]=1;
11.    for(int i=n-1;i;i--)a[i]=(a[i+1]+i)*2;
12.    上面的find函数是用函数递归写的
13.    这里是拿循环写的（应该代码是对的）
14.    几乎所有循环都能改成递归，几乎所有递归都能改成循环
15.    二者各有优劣
16. */
17. int main()

```

```

18. {
19.     scanf("%d",&n);
20.     if(n==1)printf("The monkey got %d peach in first day.\n",find(1));

21.     else printf("The monkey got %d peaches in first day.\n",find(1));
22. }

```

要想知道第一天的答案，可以通过第二天的答案得到

$$a_1 = (a_2 + 1) * 2$$

要想知道第二天的答案，可以通过第三天的答案得到

$$a_2 = (a_3 + 2) * 2$$

要想知道第三天的答案，可以通过第四天的答案得到

$$a_3 = (a_4 + 3) * 2$$

要想知道第n-1天的答案，可以通过第n天的答案得到

$$a_{n-1} = (a_n + n - 1) * 2$$

$a_n = 1$ 是已经知道的

所以能算出第n-1天的答案

所以能算出第n-2天的答案

所以能算出第n-3天的答案

所以能算出第三天的答案

所以能算出第二天的答案

所以能算出第一天的答案

所以可以通过循环 i表示第i天 a_i 表示第i天的答案

然后循环从n到1进行, $a_i = (a_{i+1} + i) * 2$ 每次通过第i+1天的答案算出第i天的答案

直到算出第1天的答案

```

int find(int x)
{
if(x==n)return 1;
else return (find(x+1)+1)*2
}

```

find(x)函数的功能是算第x天有多少果子 返回值即第x天的答案

它可以通过第x+1天的答案算出来，第x+1天的答案可以用find (x+1) 去算

T21三做一年级算术题

```

1. #include<stdio.h>
2. #include<string.h>
3. char s[100000],f;
4. int mx=0,mn=100,x;
5. int max(int a,int b){return a>b?a:b;}
6. int min(int a,int b){return a<b?a:b;}
7. int main()
8. {
9.     scanf("%s",s+1);
10.    for(int L=1;s[L];L++)if(s[L]>='0'&&s[L]<='9')x=s[L]-'0',mx=max(x,m
x),mn=min(mn,x);
11.    f=s[strlen(s+1)-2];
12.    if(f=='+')printf("%d+%d=%d\n",mx,mn,mx+mn);
13.    else if(f=='-')printf("%d-%d=%d\n",mx,mn,mx-mn);
14.    else if(f=='*')printf("%d*%d=%d\n",mx,mn,mx*mn);
15.    else if(f=='/'){if(mn==0)printf("Error!\n");else
printf("%d/%d=%d\n",mx,mn,mx/mn);}
16.    else{if(mn==0)printf("Error!\n");else printf("%d%%d=%d\n",mx,mn,mx%
mn);}
17. }

```

T22求最小的m值

```

1. #include<stdio.h>
2. int n,res;
3. int main()
4. {
5.     scanf("%d",&n);
6.     for(int i=1;;i++)
7.     {
8.         res=res+i;
9.         if(res>=n){printf("%d\n",i);break;}
10.    }
11. }

```

图形题

这几个题都相当没意思

T23数字正方形

```

1. #include<stdio.h>
2. int n,res;

```

```

3. int main()
4. {
5.     scanf("%d",&n);
6.     for(int i=1;i<=n;i++,printf("\n"))
7.         for(int j=1;j<=n;j++)
8.             if(res>=9)printf(" %d",++res);else printf("  %d",++res);
9. }

```

T24数字菱形

```

1. #include "stdio.h"
2. int main()
3. {
4.     int n,i,j;
5.     scanf("%d",&n);
6.     for(i=1;i<=n;i++)
7.     {
8.         for(j=1;j<=2*(n-i);j++)printf(" ");
9.         for(j=1;j<=i;j++)if(n-j+1<10)printf("%d ",n-j+1);else printf("%
d",n-j+1);
10.        for(j=i-1;j>=1;j--)if(n-j+1<10)printf("%d ",n-j+1);else
printf("%d",n-j+1);
11.        printf("\n");
12.    }
13.    for(i=n-1;i>=1;i--)
14.    {
15.        for(j=1;j<=2*(n-i);j++)printf(" ");
16.        for(j=1;j<=i;j++)if(n-j+1<10)printf("%d ",n-j+1);else printf("%
d",n-j+1);
17.        for(j=i-1;j>=1;j--)if(n-j+1<10)printf("%d ",n-j+1);else
printf("%d",n-j+1);
18.        printf("\n");
19.    }
20. }

```

T25倒三角形

```

1. #include<stdio.h>
2. int main()
3. {
4.     int n,i,j,k;i=j=k=1;
5.     scanf("%d",&n);
6.     for(i=1;i<=(2*n-1);i++)printf("*");
7.     printf("\n");
8.     for(j=1;j<=n-1;j++)

```

```

9.      {
10.         for (k=1;k<2*n-1;k++)
11.         {
12.             if (k==j+1||k==2*n-1-j)
13.                 printf("*");
14.             else
15.                 printf(" ");
16.             if(k==2*n-1-j)break;
17.         }
18.         printf("\n");
19.     }
20.     return 0;
21. }

```

T26计算sum的值

```

1. #include<stdio.h>
2. int n;double res;
3. int main()
4. {
5.     scanf("%d",&n);
6.     for(int i=1;i<=n;i++)res+=(1/(i*1.00));
7.     printf("sum=%.6lf\n",res);
8. }

```

T27 寻找特殊偶数

```

1. #include<stdio.h>
2. #include<string.h>
3. int x,y,res,cnt=0;
4. int a[100];
5. int cal(int n)
6. {
7.     memset(a,0,sizeof(a)); //这句话表示把a数组内的元素都改为0
8.     while(n)
9.     {
10.         if(a[n%10])return 0;
11.         a[n%10]++;n=n/10;
12.     }return 1;
13. }
14. int main()
15. {
16.     while(1)
17.     {
18.         scanf("%d%d",&x,&y);

```

```

19.     res=0;if(x==0&&y==0)break;
20.     if(x>y){printf("Error\n");continue;}
21.     if(x<=999||y>9999){printf("Error\n");continue;}
22.     for(int i=x;i<=y;i++)if((i%2==0)&&cal(i))printf("%d  ",i),res++;
23.     printf("\ncounter=%d\n",res);
24.     }
25. }

```

T28.寻找阿姆斯特朗数

```

1. #include<stdio.h>
2. #include<string.h>
3. int n,m;
4. int cal(int x,int y){int res=1;for(int i=1;i<=y;i++)res=res*x;return res;}
5. int can(int x)
6. {
7.     int res=0,y=x;
8.     while(x)res=res+cal(x%10,n),x=x/10;
9.     if(res==y)return 1;
10.    return 0;
11. }
12. int main()
13. {
14.     scanf("%d",&n);
15.     for(int i=cal(10,n-1);i<=cal(10,n)-1;i++)if(can(i))printf("%d\n",i),
        m=1;
16.     if(!m)printf("No output.\n");
17. }

```

T29 零钱换整钱*

```

1. #include<stdio.h>
2. #include<string.h>
3. int n,m;
4. int main()
5. {
6.     scanf("%d",&n);
7.     for(int i=1;;i++)
8.     {
9.         int res=1*10*i+5*12*i+10*9*i;
10.        if(res==n*100){printf("%d,%d,%d\n",10*i,12*i,9*i);return 0;}
11.        if(res>n*100)break;
12.    }printf("No change.\n");
13. }

```

T30计算天数

```
1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31};
3. int cal(int n,int m,int k,int sum)
4. {
5.     for(int i=1900;i<n;i++)if((i%4==0)&&(i%100!=0)||
        (i%400==0))sum+=366;else sum+=365;
6.     if((n%4==0)&&(n%100!=0)|| (n%400==0))a[2]=29;
7.     for(int i=1;i<=m-1;i++)sum+=a[i];sum+=k;a[2]=28;
8.     return sum;
9. }
10. int main()
11. {
12.     int n,m,k;
13.     scanf("%d%d%d",&n,&m,&k);
14.     int sum1=cal(n,m,k,0);
15.     scanf("%d%d%d",&n,&m,&k);
16.     int sum2=cal(n,m,k,0)-sum1;
17.     if(sum2==1)printf("%d day\n");
18.     else printf("%d days\n",sum2);
19. }
```

T31 计算最后三位数

```
1. #include<stdio.h>
2. int a,n,m,res=1;
3.
4. int main()
5. {
6.     scanf("%d%d",&a,&n);
7.     for(int i=1;i<=n;i++)
8.     {
9.         res=res*a;
10.        if(res>10000)m=1;
11.        if(m==1)res=res%1000;
12.    }
13.    if(m)printf("The last 3 numbers is %03d.\n",res);
14.    else printf("The last 3 numbers is %d.\n",res);
15. }
```

T32 孪生素数

```
1. #include<stdio.h>
2. int a,n,m,res=1;
3. int can(int x){for(int i=2;i*i<=x;i++)if(x%i==0)return 0;return 1;}
4. int main()
5. {
6.     scanf("%d%d",&a,&n);
7.     for(int i=a;i<=n-2;i++)
8.         if(can(i)&&can(i+2))printf("%d,%d\n",i,i+2);
9. }
```

T33计算后续日期

```
1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},n,m,k,sum,T;
3. int can(int x){if((x%4==0&&x%100!=0)|| (x%400==0))return 1;return 0;}
4. int main()
5. {
6.     scanf("%d%d%d%d",&n,&m,&k,&T);
7.     while(T-->0)
8.     {
9.         if(can(n))a[2]=29;else a[2]=28;
10.        k++;
11.        if(k>a[m])m++,k=1;
12.        if(m>12)n++,m=1;
13.        printf("%d.%d.%d\n",n,m,k);
14.    }
```

T34哥俩好

```
1. #include<stdio.h>
2. int n,a[100];
3. int main()
4. {
5.     scanf("%d",&n);
6.     for(int i=1;i<=n;i++)scanf("%d",&a[i]);
7.     for(int i=1;i<=(n+1)/2;i++)
8.         if(i!=n-i+1)printf("%d ",a[i]*a[n-i+1]);
9.     else printf("%d ",a[i]);
10.    printf("\n");
11. }
```


T35北理工的恶龙

这道题首先需要一个贪心策略就是 对于每一个恶龙 去用身高不小于它的并且身高最矮的一个勇士去杀死它

所以把恶龙 和 勇士 排序一遍就可以了

至于排序

快速排序

桶排序

冒泡排序

选择排序

堆排序

归并排序

希尔排序

以上排序算法都可以

我用的是c语言自带的快速排序

```
1. #include<stdio.h>
2. #include<stdlib.h>
3. int n,m,a[1010],b[1010],ans;
4. int cmp(const void*a,const void*b){return *(int*)a-*(int*)b;} //自定义
   比较函数
5. //如果不懂指针 可以背过它 或者自己手写其他排序函数
6. int main()
7. {
8.
9.     scanf("%d%d",&n,&m);
10.    for(int i=1;i<=n;i++)scanf("%d",&a[i]);
11.    for(int i=1;i<=m;i++)scanf("%d",&b[i]);
12.    qsort(a,n+1,sizeof(int),cmp); //调用排序函数
13.    qsort(b,m+1,sizeof(int),cmp); //调用排序函数
14.    for(int i=n;i>=1;i--)
15.    {
16.        int fla=0;
17.        for(int j=1;j<=m;j++)
18.            if(a[i]<=b[j]){fla=1;ans+=b[j];b[j]=0;break;}
19.        if(!fla){printf("bit is doomed!\n");return 0;}
20.    }printf("%d\n",ans);
21. }
```

T36谁能出线

```
1. #include<stdio.h>
2. int n,a[100],mx;
3. int max(int x,int y){return x>y?x:y;}
4. int main()
```

```

5. {
6.     for(int i=1;i<=10;i++)scanf("%d",&a[i]),mx=max(mx,a[i]);
7.     for(int i=1;i<=10;i++)if(mx==a[i])printf("%d\n",i);
8.     找到最大值 然后判断谁和最大值相同就好了
9. }

```

T37大家一起做游戏

这一类问题统称为约瑟夫环问题，网上有更优美的做法以及加强版的题目
我这里只写的模拟

```

1. #include<stdio.h>
2. int n,m,a[100],mx;
3. int max(int x,int y){return x>y?x:y;}
4. int main()
5. {
6.     scanf("%d%d",&n,&m);mx=n;
7.     int now=1,pos=1;
8.     while(mx>1)
9.     {
10.         while(pos<m)
11.         {
12.             now++;
13.             if(now>n)now=1;
14.             if(!a[now])pos++;
15.
16.         }a[now]=1;mx--,pos=0;
17.     }
18.     for(int i=1;i<=n;i++)if(!a[i])printf("The left child is NO %d.\n",
19.     i);

```

T38等值数列段

```

1. #include<stdio.h>
2. int n,m,a[100],mx,l,r;
3. int max(int x,int y){return x>y?x:y;}
4. int main()
5. {
6.     scanf("%d",&n);
7.     for(int i=1;i<=n;i++)scanf("%d",&a[i]);a[n+1]=10000000;
8.     int now=1;
9.     l为最长数列段的左端点 r为最长数列段的右端点
10.    now表示当前数列段的左端点
11.    for(int i=2;i<=n+1;i++)

```

```

12.         if(a[i]!=a[i-1])
13.         {
14.             if(mx<i-1-now+1)
15.                 l=now,r=i-1,mx=i-now;
16.             now=i;
17.         }
18.         if(mx>1)printf("The longest equal number list is from %d to %d.\n",
        l-1,r-1);
19.         else printf("No equal number list.\n");
20.     }

```

T39

```

1. #include<stdio.h>
2. int n,m,a[100][100],flag,x,y;
3. int can1(int x1,int y1,int L)
4. {
5.     for(int i=1;i<=m;i++)
6.         if(a[L][i]>a[x1][y1])return 0;
7.     return 1;
8. }
9. int can2(int x1,int y1,int R)
10. {
11.     for(int i=1;i<=n;i++)
12.         if(a[i][R]<a[x1][y1])return 0;
13.     return 1;
14. }
15. int main()
16. {
17.     scanf("%d%d",&n,&m);
18.     for(int i=1;i<=n;i++)for(int j=1;j<=m;j++)scanf("%d",&a[i][j]);
19.     for(int i=1;i<=n;i++)
20.         for(int j=1;j<=m;j++)
21.             if(can1(i,j,i)&&can2(i,j,j))flag++,x=i,y=j;
22.     if(flag)printf("Point:a[%d][%d]==%d\n",x-1,y-1,a[x][y]);
23.     else printf("No Point\n");
24. }
25. 枚举每个点 判断它是不是符合条件就好了

```

复杂度

值此这个题 写（复制）一下 复杂度问题
以后也不一定会用到

算法复杂度分为时间复杂度和空间复杂度。

其作用：

时间复杂度是指执行算法所需要的计算工作量；

而空间复杂度是指执行这个算法所需要的内存空间。

（算法的复杂性体现在运行该算法时的计算机所需资源的多少上，计算机资源最重要的是时间和空间（即寄存器）资源，因此复杂度分为时间和空间复杂度）。

简单来说，时间复杂度指的是语句执行次数，空间复杂度指的是算法所占的存储空间

时间复杂度

计算时间复杂度的方法：

用常数1代替运行时间中的所有加法常数

修改后的运行次数函数中，只保留最高阶项

去除最高阶项的系数

按数量级递增排列，常见的时间复杂度有：

常数阶 $O(1)$ ，对数阶 $O(\log 2n)$ ，线性阶 $O(n)$ ，

线性对数阶 $O(n\log 2n)$ ，平方阶 $O(n^2)$ ，立方阶 $O(n^3)$ ，...

k次方阶 $O(n^k)$ ，指数阶 $O(2^n)$ 。

在常见的算法中如果复杂度为log 一般会写底数 默认底数为2

原因：

1 很多算法确实以2为底 比如二分 线段树 树状数组 归并排序等等

2 即使不是以2为底 也可以通过换底公式 改成以2为底 然后忽略系数

随着问题规模n的不断增大，上述时间复杂度不断增大，算法的执行效率越低

所以代码的时间复杂度越低越好

最坏时间复杂度和平均时间复杂度

最坏情况下的时间复杂度称最坏时间复杂度。一般不特别说明，讨论的时间复杂度均是最坏情况下的时间复杂度。

这样做的原因是：最坏情况下的时间复杂度是算法在任何输入实例上运行时间的上界，这就保证了算法的运行时间不会比任何更长。

平均时间复杂度是指所有可能的输入实例均以等概率出现的情况下，算法的期望运行时间。设每种情况的出现的概率为 p_i ，平均时间复杂度则为 $\sum(p_i * f(n))$

空间复杂度

空间复杂度(Space Complexity)是对一个算法在运行过程中临时占用存储空间大小的量度，记做 $S(n)=O(f(n))$ 。

对于一个算法来说，空间复杂度和时间复杂度往往是相互影响的。当追求一个较好的时间复杂度时，可能会使空间复杂度的性能变差，即可能导致占用较多的存储空间；反之，当追求一个较好的空间复杂度时，可能会使时间复杂度的性能变差，即可能导致占用较长的运行时间。

当然也会有拿时间换空间的例子 不过我接触到的还是拿空间换时间的多

作者：Eve慕

来源：CSDN

原文：<https://blog.csdn.net/zxm490484080/article/details/72210501>

有兴趣的话可以去网上找更多博客资料 以及 去算一下之前and以后写的题的时空复杂度

T40视频

计算公式 见代码

为啥是这个计算公式 我就知道了 233

我是用结果反推过来的

因为这个题可以用c++交 我就写的c++了

数据爆不了 long long 但有可能在计算过程中爆int

解决办法:

1手写高精度

2用c++ long long类型 或者其他语言的大数类型

3在计算乘法过程中也做除法

一共要除 2^{13} 如果可以除以 2^x 就先除掉 只要保证一共除了 2^{13} 就可以了

ps:如果以后想学c++ 记住这个库名哦 它包括了很多很多库 所以其他库名几乎就不用打了

```
1. #include<bits/stdc++.h>
2. int main()
3. {
4.     long long n,m,s;
5.     scanf("%lld*%lld%lld",&n,&m,&s);
6.     long long ans=n*m*43200LL;
7.     ans/=8;
8.     ans/=1024;
9.     ans*=s;
10.    printf("%lld\n",ans);
11. }
```

T41音频

与上题类似

```
1. #include<bits/stdc++.h>
2. int main()
3. {
4.     long long n,m,s;
5.     scanf("%lldhz%lld%lld",&n,&m,&s);
6.     long long ans=n*m*120LL;
7.     ans/=8;
8.     ans/=1024;
9.     ans*=s;
10.    printf("%lld\n",ans);
```

```
11. }
```

T42 汉字储存

计算方式见代码

```
1. #include<bits/stdc++.h>
2. int main()
3. {
4.     long long n,m,s;
5.     scanf("%lld%lld",&n,&m);
6.     printf("%lld\n",n*n*m/8);
7. }
```

T43 n位正整数的个数

这种题来学递归搜索最好了

```
1. #include<stdio.h>
2. int ans,n;
3. void dfs(int x,int step)
4. {
5.     if(step==n){ans++;return;}
6.     for(int i=0;i<=9;i++)
7.     {
8.         if(step==0&&i==0)continue;
9.         if((x*10+i)%((step+1)*(step+1))==0)
10.             dfs(x*10+i,step+1);
11.     }
12. }
13. int main()
14. {
15.     scanf("%d",&n);
16.     if(n!=0)dfs(0,0);
17.     printf("%d\n",ans);
18. }
```

T44 合并字符串

把两个已经有序序列合并 只需要判断两个序列的当前序列头就好了 这也是归并排序的思想之一 当然这个题还有更优美的做法 比如用桶排序 代码会更简洁

<https://www.cnblogs.com/chengxiao/p/6194356.html> 这是归并排序的博客

<https://blog.csdn.net/YinhJiang/article/details/80397415> 这是桶排序的博客

```
1. #include<stdio.h>
2. #include<string.h>
3. int n,m;
4. char s1[500],s2[500];
5. int main()
6. {
7.     scanf("%s",s1+1);n=strlen(s1+1);
8.     scanf("%s",s2+1);m=strlen(s2+1);
9.     int i=1,j=1;
10.    while(i<=n&&j<=m)
11.    {
12.        if(s1[i]<s2[j])printf("%c",s1[i]),i++;
13.        else printf("%c",s2[j]),j++;
14.    }
15.    while(i<=n)printf("%c",s1[i]),i++;
16.    while(j<=m)printf("%c",s2[j]),j++;
17.    printf("\n");
18. }
```

T45 删除重复字符

开个数组 记录该字符有没有出现过就好了

```
1. #include<stdio.h>
2. char s[1000];
3. int a[1000];
4. int main()
5. {
6.     scanf("%s",s+1);
7.     for(int i=1;s[i];i++)
8.         if(a[(int)(s[i])]==0)printf("%c",s[i]),a[(int)(s[i])]=1;
9.     printf("\n");
10. }
```

T46 组成最大数

```
1. #include<stdio.h>
2. #include<string.h>
3. int a[10];
4. int main()
5. {
```

```

6.     char ch=getchar();
7.         while(ch>='0'&&ch<='9'){a[ch-'0']++;ch=getchar();}
8.     for(int i=9;i>=0;i--)while(a[i]--)printf("%d",i);
9.     printf("\n");
10. }

```

T47

```

1. #include<stdio.h>
2. #include<string.h>
3. char s[5000],s1[100],s2[5050];
4. int cal()
5. {
6.     int rs=0,flag=1;
7.     for(int i=2;s1[i];i++)
8.     {
9.         if(s1[i]=='-')flag=-1;
10.        else rs=rs*10+s1[i]-'0';
11.    }return rs*flag;
12. }
13. int res,cnt,sum;
14. int main()
15. {
16.     int n;
17.     scanf("%s%s",s+1,s1+1);
18.     if(strlen(s1+1)==1)scanf("%d",&n);
19.     else n=cal();
20.     for(int i=1;s[i];i++)if(s[i]=='.')res=i-1+n;else s2[++cnt]=s[i];
21.     if(!res)res=strlen(s+1)+n;
22.     if(res<1)
23.     {
24.         printf("0.");res++;while(res<1&&sum<8)printf("0"),res++,sum++;
25.
26.         for(int i=1;i<=8-sum;i++)printf("%c",s2[i]?s2[i]:'0');
27.     }
28.     else
29.     {
30.         for(int i=1;i<=res;i++)printf("%c",s2[i]?
31.         s2[i]:'0');printf(".");
32.         for(int i=res+1;i<=res+8;i++)printf("%c",s2[i]?s2[i]:'0');
33.     }printf("\n");
34. }

```

T48


```

1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},y,cnt;
3. int ans[100];
4. int cal(int n,int m,int k)
5. {
6.     a[2]=28;int sum=0;
7.     for(int i=1900;i<n;i++)if((i%4==0)&&(i%100!=0)||
    (i%400==0))sum+=366;else sum+=365;
8.     if((n%4==0)&&(n%100!=0)|| (n%400==0))a[2]=29;
9.     for(int i=1;i<m;i++)sum+=a[i];sum+=k;
10.    return (sum)%7;
11. }
12. int main()
13. {
14.     scanf("%d",&y);
15.     for(int i=1;i<=12;i++)if(cal(y,i,13)==5)ans[++cnt]=i;
16.     if(cnt>1)printf("There are %d Black Fridays in year %d.\n",cnt,y);
17.     else printf("There is %d Black Friday in year %d.\n",cnt,y);
18.     if(cnt)
19.     {
20.         if(cnt!=1)printf("They are:\n");else printf("It is:\n");
21.         for(int i=1;i<=cnt;i++)printf("%d/%d/13\n",y,ans[i]);
22.     }
23. }

```

T49再算天数

只用提交一个主函数就好了

```

1. int main()
2. {
3.     int n,m,k;
4.     scanf("%d%d%d",&n,&m,&k);
5.     int sum1=0,sum2=0;
6.     for(int i=1900;i<n;i++)sum1+=year_days(i);
7.     sum1+=days(n,m,k);
8.     scanf("%d%d%d",&n,&m,&k);
9.     for(int i=1900;i<n;i++)sum2+=year_days(i);
10.    sum2+=days(n,m,k);
11.    if(sum2-sum1==1)printf("1 day\n");
12.    else printf("%d days\n",sum2-sum1);
13. }

```

细心的dalao一定已经发现没有47 48题了 因为突然发现我还没做233

T50 求最大公约数

将高中学的欧几里得算法 用递归写一下就好了

安利c++中有__gcd()函数 可以直接用

而且受bit/stdc++.h的影响 我忘记这个函数在哪个库了233

```
1. #include<stdio.h>
2. #include<string.h>
3. int n,a[100],flag,m;
4. int gcd(int a,int b)
5. {
6.     if(a%b==0) return b;
7.     return gcd(b,a%b);
8. }
9. int main()
10. {
11.     scanf("%d%d",&n,&m);
12.     printf("%d\n",gcd(n,m));
13. }
```

T51 求序列和

```
1. #include<stdio.h>
2. #include<string.h>
3. int n,m;
4. char s1[500],s2[500];
5. double find(int x,double y)
6. {
7.     if(x==n) return y/(x*1.00);
8.     else return find(x+1,-y)+y/(x*1.00);
9. }
10. int main()
11. {
12.     scanf("%d",&n);
13.     double res=0;
14.     if(n==1) printf("1\n");
15.     else printf("%.6lf\n",1.00+find(2,1));
16. }
```

T52 函数的递归

按照给定的递推公式就好了

- 1 压行真爽
- 2 别人代码怎么这么短 肯定刻意压行 瞧不起人类定律

```
1. int findf(int x)
2. {
3.     if(x>4)return findf(x-1-x%2)+findf(x-3-x%2);
4.     else if(x>=0)return 1;else return -1;
5. }
```

T53回文字符串

- 1 递归能写的题怎么能写循环呢
- 2 循环真爽

```
1. #include<stdio.h>
2. #include<string.h>
3. char s[100000];int n,flag;
4. int main()
5. {
6.     scanf("%s",s+1);n=strlen(s+1);
7.     for(int i=1;i<=n/2;i++)if(s[i]!=s[n-i+1])flag=1;
8.     if(flag)printf("No\n");else printf("Yes\n");
9. }
```

T54 子串反转

```
1. void reverse(char str[],int start,int end)
2. {
3.     while(!((str[end]>='0'&&str[end]<='9'))end--;
4.     if(end<=start)return;
5.     char x;
6.     x=str[end];
7.     str[end]=str[start];
8.     str[start]=x;
9.     //交换当前序列的开头与结尾
10.    reverse(str,start+1,end-1);    //递归处理左端点+1 右端点-1的子序列
11. }
```

T55多少种表示方法

这个题用递归去写，也有可能可以过，，不过代码难写而且复杂度高

```
1. #include<stdio.h>
2. int n,m,f[1010];
3. int main()
4. {
5.     scanf("%d%d",&m,&n);
6.     f[0]=1;
7.     for(int i=1;i<=n;i++) for(int j=i;j<=m;j++) f[j]+=f[j-i];
8.     printf("%d\n",f[m]);
9. }
```

$f[j]$ 表示 m 可表示为一些不超过 n 的整数之和，表示出 j 方式的数目

$$f[j] += f[j - i], i \leq \min(j, n)$$

T56 单词排序

```
1. #include<stdio.h>
2. #include<string.h>
3. char s[10][1000];int n,flag;
4. int max(int x,int y){return x>y?x:y;}
5. int pan(int x,int y) //判断大小函数
6. {
7.     int l1=strlen(s[x]+1),l2=strlen(s[y]+1),d=max(l1,l2);
8.     for(int i=1;i<=d;i++)if(s[x][i]<s[y][i])return 1; else if(s[x][i]>s
9.     [y][i])return 0;
10.    return 0;
11. }
12. void sw(int x,int y) //交换函数
13. {
14.     char t;
15.     int l1=strlen(s[x]+1),l2=strlen(s[y]+1),d=max(l1,l2);
16.     for(int i=1;i<=d;i++)t=s[x][i],s[x][i]=s[y][i],s[y][i]=t;
17. }
18. int main()
19. {
20.     for(int i=1;i<=5;i++)scanf("%s",s[i]+1);
21.     for(int i=1;i<=5;i++)for(int j=1;j<=4;j++)if(pan(j,j+1))sw(j,j+1);
22.     for(int i=1;i<=5;i++)printf("%s\n",s[i]+1);
23. }
```

T57

```

1. #include<stdio.h>
2. #include<string.h>
3. int n,a[100],flag;
4. void cal(int x){for(int i=1;i<=3;i++)a[x%10]++,x=x/10;}
5. int can(){if(a[0])return 0;for(int i=1;i<=9;i++)if(a[i]>1)return 0;return 1;}
6. int main()
7. {
8.     scanf("%d",&n);
9.     for(int i=n*100;i<=(n+1)*100-1;i++) //暴力枚举然后判断就好了
10.    {
11.        if(i*2>999||i*3>999)break;
12.        memset(a,0,sizeof(a));
13.        cal(i),cal(i*2),cal(i*3);
14.        if(can())printf("%d,%d,%d\n",i,i*2,i*3),flag=1;
15.    }
16.    if(!flag)printf("0,0,0\n");
17. }

```

T58

算给定日期距离2012年4月9日多少周，然后看该禁止哪个车就好了

```

1. #include<stdio.h>
2. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31};
3. int cal(int n,int m,int k,int sum)
4. {
5.     for(int i=1900;i<n;i++)if((i%4==0)&&(i%100!=0)||
6.     (i%400==0))sum+=366;else sum+=365;
7.     if((n%4==0)&&(n%100!=0)|| (n%400==0))a[2]=29;
8.     for(int i=1;i<=m-1;i++)sum+=a[i];sum+=k;a[2]=28;
9.     return sum;
10. }
11. int X[10][10],Y[10][10];
12. int main()
13. {
14.     int n,m,k;
15.     n=2012,m=4,k=9;
16.     int sum1=cal(n,m,k,0);
17.     scanf("%d%d%d",&n,&m,&k);
18.     int sum2=cal(n,m,k,0)-sum1;
19.     int day=(1+sum2)%7;
20.     if(day==6||day==0){printf("Free.\n");return 0;}
21.     sum2++;
22.     X[1][1]=3,X[1][2]=4,X[1][3]=5,X[1][4]=1,X[1][5]=2;
23.     int week=(sum2/7)+(sum2%7!=0),res=0;
24.     for(int i=1;;i++)
25.         if((i-1)*13<=week&&(i)*13>=week)

```

```

25.         {res=i;break;}
26.     int ans1=(X[1][day]-(res-1));
27.     ans1=(ans1%5+5)%5;
28.     if(ans1==0)ans1=5;
29.     int ans2=(ans1+5)%10;
30.     printf("%d and %d.\n",ans1,ans2);
31. }

```

01.c (D:\program file\code) - GVIM

```

1 #include<stdio.h>
2 int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31};
3 int cal(int n,int m,int k,int sum)
4 {
5     for(int i=1900;i<n;i++)if((i%4==0)&&(i%100!=0)|| (i%400==0))sum+=366;else sum+=365;
6     if((n%4==0)&&(n%100!=0)|| (n%400==0))a[2]=29;
7     for(int i=1;i<=m-1;i++)sum+=a[i];sum+=k;a[2]=28;
8     return sum;
9 }
10 int X[10][10],Y[10][10];
11 int main()
12 {
13     int n,m,k;
14     n=2012,m=4,k=9;
15     int sum1=cal(n,m,k,0);
16     scanf("%d%d%d",&n,&m,&k);
17     int sum2=cal(n,m,k,0)-sum1;
18     int day=(1+sum2)%7;
19     if(day==6||day==0){printf("Free.\n");return 0;}
20     sum2++;
21     X[1][1]=3,X[1][2]=4,X[1][3]=5,X[1][4]=1,X[1][5]=2;
22     int week=(sum2/7)+(sum2%7!=0),res=0;
23     for(int i=1;i++)
24         if((i-1)*13<=week&&(i)*13>=week)
25             {res=i;break;}
26     int ans1=(X[1][day]-(res-1));
27     ans1=(ans1%5+5)%5;
28     if(ans1==0)ans1=5;
29     int ans2=(ans1+5)%10;
30     printf("%d and %d.\n",ans1,ans2);
31 }

```

T59

把之前字符串存起来，如果重复出现则输出序号 否则输出字符串

(ps 文件有10m 程序只给1m，说明老师非常良心的没有构造数据去卡这种做法)

(这种做法理论是可以被卡掉的)

```

1. #include<stdio.h>
2. #include<string.h>
3. char s[90],b[50],a[7000][30];int sum;
4. void sol(int l,int r){
5.     for(int i=0;b[i];i++)b[i]='\0';
6.     for(int i=l;i<=r;i++)b[i-l]=s[i];int flag=0;
7.     for(int i=1;i<=sum;i++)if(strcmp(a[i],b)==0)
8.         {printf("%d",i);flag=1;break;}
9.     if(!flag){printf("%s",b);if(b[0]!='\0')strcpy(a[++sum],b);}
10. }
11. int main()
12. {
13.     while(gets(s+1)!=NULL) {
14.         for(int i=1,j=1;s[i];i=j+1,j++){
15.             if((s[i]>='a'&& s[i]<='z')||(s[i]>='A'&& s[i]<='Z')){

```

```

15.         for(;s[j];j++)if(!((s[j]>='a'&& s[j]<='z')||(s[j]>='A'&& s[j]
    <='Z'))))break;
16.         j--;sol(i,j);
17.     }else printf("%c",s[i]);
18.     }printf("\n");
19. }
20. }

```

T60

看脸的一个题

注意加空循环

`srand((int)time(NULL));`

表示用当前时间作为随机数种子然后去得到随机数据

为啥要加空循环，是让程序运行有间隔 不然当前时间相同，种子相同，随机出来的数据也相同

```

1. #include <stdlib.h>
2. #include<stdio.h>
3. #include<time.h>
4. int main()
5. {
6.     int n=0;
7.     for(int j=1;j<=20;j++)for(int i=1;i<=100000000;i++)n+=i;
8.     srand((int)time(NULL));
9.     n=rand()%3;
10.    if(n<=1)printf("2018\n");
11.    else printf("2019\n");
12. }

```

T61

```

1. #include<stdio.h>
2. int ans,n;
3. char s[3000],x;
4. int main()
5. {
6.     FILE *fp;
7.     scanf("%s",s);
8.     fp=fopen(s,"r",stdin);
9.     x=fgetc(fp);
10.    while(x!=EOF)printf("%c",x),x=fgetc(fp);;
11. }

```

T62

```
1. #include<stdio.h>
2. #include<string.h>
3. int ans,n,cnt;
4. char s[3000],x;
5. struct node
6. {
7.     char id[10],name[10],sex,day[20];
8.     int y,nid;
9. }e[3000],no[3000],t;
10. void swap(int x,int y)
11. {
12.     t=no[x];
13.     no[x]=no[y];
14.     no[y]=t;
15. }
16. int main()
17. {
18.     for(int i=1;i<=180;i++)
19.     {
20.         scanf("%s",s+1);e[i].nid=0;
21.         for(int j=1;j<=4;j++)e[i].id[j]=s[j],e[i].nid=e[i].nid*10+s[j]-
22.         '0';
23.         for(int j=6;j<=15;j++)e[i].day[j-5]=s[j];
24.         for(int j=1;j<=10;j++)
25.             if(e[i].day[j]!='-')
26.                 e[i].y=e[i].y*10+e[i].day[j]-'0';
27.         e[i].sex=s[17];
28.         for(int j=19;j<=24;j++)if(s[j])e[i].name[j-18]=s[j];
29.         if(e[i].y<=20130901)no[++cnt]=e[i];
30.     }
31.     for(int i=1;i<cnt;i++)
32.     {
33.         for(int j=1;j<cnt;j++)
34.             if(no[j].nid>no[j+1].nid)
35.                 swap(j,j+1);
36.     }
37.     printf(" BH    BirthDay  Name  Sex\n");
38.     printf("-----\n");
39.     for(int i=1;i<=cnt;i++)
40.     {
41.         for(int j=1;j<=4;j++)printf("%c",no[i].id[j]);printf(" ");
42.         for(int j=1;j<=10;j++)printf("%c",no[i].day[j]);printf(" ");
43.         for(int j=1;j<=6;j++)if(j<=strlen(no[i].name+1))printf("%c",no[i].name[j]);else
44.             printf(" ");printf(" ");
45.         printf("%c\n",no[i].sex);
46.     }
```


T63

```
1. #include<stdio.h>
2. #include<string.h>
3. int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},n,m,k,sum,T,d,Nn,Nm,
   Nk;
4. int can(int x){if((x%4==0&& x%100!=0)|| (x%400==0))return 1;return 0;}
5. char s[100];
6. int main()
7. {
8.     scanf("%s",s+1);
9.     scanf("%d",&T);
10.    printf("[2017-01-01~");printf("%s",s+1);printf(",%d]\n",T);
11.    n=2017,m=1,k=1,d=7;
12.    if(strlen(s+1)!=10){printf("ERROR!\n");return 0;}
13.    for(int i=1;i<=4;i++)Nn=Nn*10+s[i]-'0';
14.    for(int i=6;i<=7;i++)Nm=Nm*10+s[i]-'0';
15.    for(int i=9;i<=10;i++)Nk=Nk*10+s[i]-'0';
16.    if((2018*10000+12*100+31)<(Nn*10000+Nm*100+Nk)|| (Nn*10000+Nm*100+N
   k)<0){printf("ERROR!\n");return 0;}
17.    if((n*10000+m*100+k)>(Nn*10000+Nm*100+Nk)){printf("ERROR!\n");retur
   n 0;}
18.    if(can(Nn))a[2]=29;
19.    if(Nk>a[Nm]|| Nk<1|| Nm>13|| Nm<1){printf("ERROR!\n");return 0;}
20.    while(n!=Nn|| m!=Nm|| k!=Nk)
21.    {
22.        if(can(n))a[2]=29;else a[2]=28;
23.        k++;d++;
24.        if(k>a[m])m++,k=1;
25.        if(m>12)n++,m=1;
26.        if(d>7)d-=7;
27.        if(d==T)printf("%03d %d-%02d-%02d\n",++sum,n,m,k);
28.    }if(!sum)printf("NO DATE FOUND!\n");
29. }
```

```

1 #include<stdio.h>
2 #include<string.h>
3 int a[100]={0,31,28,31,30,31,30,31,31,30,31,30,31},n,m,k,sum,T,d,Nn,Nm,Nk;
4 int can(int x){if((x%4==0&&x%100!=0)|| (x%400==0))return 1;return 0;}
5 char s[100];
6 int main()
7 {
8     scanf("%s",s+1);
9     scanf("%d",&T);
10    printf("[2017-01-01~");printf("%s",s+1);printf(",%d]\n",T);
11    n=2017,m=1,k=1,d=7;
12    if(strlen(s+1)!=10){printf("ERROR!\n");return 0;}
13    for(int i=1;i<=4;i++)Nn=Nn*10+s[i]-'0';
14    for(int i=6;i<=7;i++)Nm=Nm*10+s[i]-'0';
15    for(int i=9;i<=10;i++)Nk=Nk*10+s[i]-'0';
16    if((2018*10000+12*100+31)<(Nn*10000+Nm*100+Nk)|| (Nn*10000+Nm*100+Nk)<0){printf("ERROR!\n");return 0;}
17    if((n*10000+m*100+k)>(Nn*10000+Nm*100+Nk)){printf("ERROR!\n");return 0;}
18    if(can(Nn))a[2]=29;
19    if(Nk>a[Nm]||Nk<1||Nm>13||Nm<1){printf("ERROR!\n");return 0;}
20    while(n!=Nn||m!=Nm||k!=Nk)
21    {
22        if(can(n))a[2]=29;else a[2]=28;
23        k++;d++;
24        if(k>a[m])m++,k=1;
25        if(m>12)n++,m=1;
26        if(d>7)d-=7;
27        if(d==T)printf("%03d %d-%02d-%02d\n",++sum,n,m,k);
28    }if(!sum)printf("NO DATE FOUND!\n");
29 }

```

T64

这题因为可以用c++写，我就直接用的c++的排序函数，没有自己再手写

```

1. #include<cstdio>
2. #include<algorithm>
3. #define ll long long
4. using namespace std;
5. char s[100];
6. struct XsStru{unsigned int ID; char Name[8]; char Sex; unsigned char Km
   1; unsigned char Km2; unsigned char Km3;int sum;    }a[400300];
7. bool cmp(XsStru b,XsStru c){if(b.sum!=c.sum)return b.sum>c.sum;else ret
   urn b.ID<c.ID;}    //自定义比较函数
8. int main()
9. {
10.     scanf("%s",s);
11.     FILE *fid;
12.     fid = fopen(s,"rb");
13.     int cnt=1;
14.     for(;;cnt++){
15.         int d=fread(&a[cnt].ID,sizeof(int),1,fid);
16.         if(d==0)break;
17.         fread(a[cnt].Name,8,1,fid);
18.         fread(&a[cnt].Sex,
   sizeof(char),1,fid);fread(&a[cnt].Km1,sizeof(char),1,fid);
19.         fread(&a[cnt].Km2, sizeof(char),1,fid);fread(&a[cnt].Km3,sizeof(c
   har),1,fid);
20.         a[cnt].sum=(int)a[cnt].Km1+(int)a[cnt].Km2+(int)a[cnt].Km3;
21.     }cnt--;
22.     fclose(fid);
23.     sort(a+1,a+1+cnt,cmp);

```

```

24.     printf("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----\n");
25.     printf("|                TOP 99 of %-6d                |\n",c
nt);
26.     printf("+----+-----+-----+----+-----+-----+-----+-----+-----\n");
27.     printf("| TOP |   ID   | Name    |Sex| SUM | KM1 | KM2 | KM3
|\n");
28.     for(int i=1;i<=99;i++)
29.     {
30.         if(i%5==1)printf("+----+-----+-----+----+-----+-----+-----\n");
31.         printf("|%3d  | %06d | %.8s | %c | %3d | %3d | %3d | %3d |
\n",i,a[i].ID,a[i].Name,a[i].Sex,a[i].sum,a[i].Km1,a[i].Km2,a[i].Km3);
32.     }printf("+----+-----+-----+----+-----+-----+-----+-----+-----\n");
33.
34. }

```

T65

这题用c语言函数应该会缩短很多代码量

```

1. #include <stdlib.h>
2. #include<stdio.h>
3. #include <string.h>
4. #include "malloc.h"
5. #define MaxPass 66 // 最多66段密文
6. #define NumbPwdTable 5 // 密码表的份数
7. #define SizePwdTable 0x210 // 每份密码表占用字节数
8. struct PassStru{unsigned int Offset; unsigned short int Size; unsigned
   char Key; char cJiangYou;}e[5050];
9. char MsgErr[]="Wrong Key!",MsgSn[]="Serial No.: ",MsgMD9[]="MD9: ";
10. char s[1000],s2[500100];
11. int s3[500100];
12. int n;
13. int can()
14. {
15.     for(int i=1;i<=10;i++)if(s3[i]!=MsgErr[i-1])return 1;
16.     return 0;
17. }
18. int can2(int left)
19. {
20.     for(int i=left;i<=left+11;i++)
21.         if(s3[i]!=MsgSn[i-left])return 0;
22.     return 1;
23. }

```

```

24. int can3(int left)
25. {
26.     for(int i=left;i<=left+4;i++)
27.         if(s3[i]!=MsgMD9[i-left])return 0;
28.     return 1;
29. }
30. int main()
31. {
32.     scanf("%s",s);scanf("%d",&n);
33.     FILE *fid;
34.     fid = fopen(s,"rb");
35.     for(int i=1;i<=5;i++)
36.     {
37.         for(int j=1;j<=66;j++)
38.         {
39.             int cnt=(i-1)*66+j;
40.             fread(&e[cnt].Offset,sizeof(unsigned int),1,fid);
41.             fread(&e[cnt].Size,sizeof(unsigned short int),1,fid);
42.             fread(&e[cnt].Key,sizeof(unsigned char),1,fid);
43.             fread(&e[cnt].cJiangYou,sizeof(char),1,fid);
44.         }
45.     }
46.     char x;int top=0;
47.     while(!feof(fid))
48.     {
49.         fread(&x,sizeof(char),1,fid);
50.         s2[++top]=x;
51.     }
52.     fclose(fid);
53.     for(int j=1;j<=5;j++)
54.     {
55.         int off=e[(j-1)*66+n].Offset-2640;
56.         int siz=e[(j-1)*66+n].Size;
57.         unsigned char add=e[(j-1)*66+n].Key;
58.         for(int i=off+1;i<=off+siz;i++)s3[i-off]=((int)s2[i]+
(int)add)%256;
59.         if(can())
60.         {
61.             printf("XH: %02d\n",n);
62.             for(int i=1;i<=siz;i++)
63.             {
64.                 if(can2(i))
65.                 {
66.                     printf("SN: ");
67.                     for(int k=i+12;k<=i+12+18;k++)printf("%c",s3[k]);
68.                     printf("\n");
69.                 }
70.                 if(can3(i))
71.                 {
72.                     printf("M9: ");
73.                     for(int k=i+5;k<=i+5+31;k++)printf("%c",s3[k]);
74.                     printf("\n");

```

```
75.         }
76.     }
77.     break;
78. }
79. }
80. }
```

T66

```
1. #include<stdio.h>
2. #include<string.h>
3. int n,m;
4. char s[100],f[10100];
5. char x1,x2,x3;
6. void print(int x)
7. {
8.     for(int i=7;i>=0;i--)
9.         if(x&(1<<i))printf("%c",x3);
10.        else printf("%c",x2);
11. }
12. int main()
13. {
14.     scanf("%d,%s",&m,s);
15.     scanf("%d,%c,%c",&n,&x2,&x3);
16.     int n1=n/100,n2=n%100;
17.     int sum=(n1-16)*94+n2;
18.     FILE *fid;
19.     fid=fopen(s,"rb");
20.     for(int i=1;i<=sum;i++)fread(f+1,(m*m)/8,1,fid);
21.     for(int i=1;i<=m;i++)
22.     {
23.         for(int k=1;k<=m/8;k++)print(f[(i-1)*(m/8)+k]);
24.         printf("\n");
25.     }
26. }
27.
```