

太戈编程
etiger.vip

信息学算法

循环控制分析

蛇形填表

输入行数和列数，输出对应的蛇形数字表格。
填表规则请参考样例。

输入样例：

3 3

输出样例：

1 2 3

6 5 4

7 8 9

输入样例：

2 4

输出样例：

1 2 3 4

8 7 6 5

代码1

如何
推理
填空

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int m,n;
5      cin>>m>>n;
6      bool flag= ;
7      for(int i=0; ;i++){
8          int tmp= ;
9          if(flag){
10             for(int j=0; ;j++)
11                 cout<<tmp+j<<" ";
12             }else{
13                 for(int j= ;j>=0;j--)
14                     cout<< <<" ";
15             }
16             ;
17             cout<<endl;
18         }
19         return 0;
20     }
```

代码1

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int m,n;
5     cin>>m>>n;
6     bool flag=
7     for(int i=0; ;i++){
8         int tmp=
9         if(flag){
10             for(int j=0; ;j++)
11                 cout<<tmp+j<<" ";
12             }else{
13                 for(int j= ;j>=0;j--)
14                     cout<< <<" ";
15             }
16             cout<<endl;
17         }
18     return 0;
19 }
20 }
```

第一步：纸和笔写所有变量含义

| | |
|--------|----------------|
| m代表 | 总行数 |
| n代表 | 总列数 |
| flag代表 | 这行是否从小到大输出 |
| i代表 | 枚举的行号 |
| j代表 | 枚举的偏差 |
| tmp代表 | 每行基准数 每行最小数 |

第二步：尝试翻译每一行写注释

| |
|------------------|
| 发现cout次数由ij 循环控制 |
| 发现endl次数由i 循环控制 |

第三步：观察变量们的生命线

| |
|-------------|
| 发现flag没有变化！ |
|-------------|

代码2

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int n,m;
5     cin>>n>>m;
6     for(int i=1; ; i++){
7         if ( ) {
8             int a= ;
9             int b= ;
10            for(int j=a; j<=b; j++)
11                cout<<j<<" ";
12        }else{
13            int a= ;
14            int b= ;
15            for(int j=a; j>=b; j--)
16                cout<<j<<" ";
17        }
18        cout<<endl;
19    }
20    return 0;
21 }
```

第一步：纸和笔写所有变量含义

| | |
|-----|----------------|
| m代表 | 总列数!! |
| n代表 | 总行数!! |
| a代表 | 循环起始数 每行最左边 |
| b代表 | 循环最终数 每行最右边 |

第二步：尝试翻译每一行写注释

j 循环次数一定是 总列数

推理步骤

1

输入变量含义

确认后写下来

输出变量含义

确认后写下来

其他变量含义

猜测后写下来

2

尝试翻译每一行写注释
能理解的一定要写翻译
不理解的先空缺

3

对每个变量
观察生命线

哪些变量应该用但没有用过
哪些变量应该变但没变过

输入变量

输出变量

判断语句
里的变量

不断重复
以上3个步骤

埃及分数

设计一个程序，把一个真分数表示为埃及分数之和的形式。所谓埃及分数，是指分子为1的形式。古代埃及有一个非常奇怪的习惯，他们喜欢把一个分数表示为若干个分子为1的分数之和的形式。如， $7/8=1/2+1/3+1/24$ 。下面是一种贪心算法，由数学家菲波那契提出的，基本思想是：

- (1) 设某个真分数的分子为A，分母为B；
- (2) 把B除以A的商的整数部分加1后的值作为埃及分数的某一个分母c；
- (3) 将A乘以C减去B作为新的A；将B乘以C作为新的B；
- (4) 如果A大于1且能整除B，则最后一个分母为B/A 否则如果A = 1，则最后一个分母为B，否则转步骤（2）。

输入样例：

5 6

输入样例：

7 8

输出样例：

$5/6=1/2+1/3$

输出样例：

$7/8=1/2+1/3+1/24$

电脑完成填空
运行测试

```

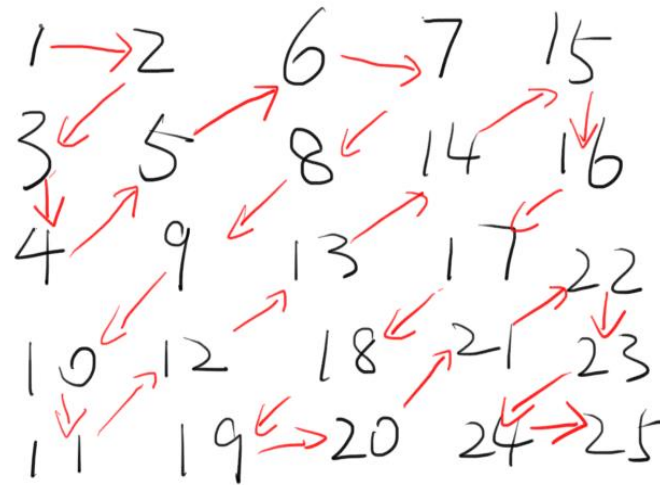
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main(){
4      int a, b, c;
5      cin >> a >> b;
6      cout << a << "/" << b << "=";
7      while (  ) {
8          c = b/a+1;
9          a = 
10         b = 
11         cout << "1/" << 
12         if(b%a==0 && a > 1){
13             cout << "+1/" << 
14             return 0;
15         }
16         if(b%a!=0){
17             cout << "+";
18         }
19     }
20     
21     return 0;
22 }
```

代码2

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main(){
4      int a, b, c;
5      cin >> a >> b;
6      cout << a << "/" << b << "=";
7      while  {
8          c = b/a+1;
9          a = a * c - b;
10         b = b * c;
11         cout << "1/" << 
12     }
13     cout << "1/" << 
14     return 0;
15 }
```

Z字填表

输入n，生成一个n*n的Z形方阵。所谓Z形方阵，请见下面的样例。下图是一个5*5的Z形方阵。



```

4  const int N=100;
5  int a[N][N];
6  int main() {
7      int n,i,j,k,t;
8      cin>>n;
9      t=
10     for(k=1;k<=n;k++) {
11         if(k%2==1)
12             for(j=1;j<=k;j++) {
13                 i =
14                 t++;
15                 a[i][j] =
16                 a[n+1-i][n+1-j] = n*n+1-t;
17             }
18         else for(j=k;j>=1;j--) {
19             i = k+1-j;
20             a[i][j] = t;
21             a[n+1-i][n+1-j] =
22         }
23     }
24 }

```

有理数

数学家康托证明了有理数（某整数除以某整数）是可枚举的。证明非常巧妙，小学生也能看懂，他用了这一张表来说明任意给定有理数都会在表格里出现。

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 |
| 2/1 | 2/2 | 2/3 | 2/4 | 2/5 | ... |
| 3/1 | 3/2 | 3/3 | 3/4 | ... | ... |
| 4/1 | 4/2 | 4/3 | ... | ... | ... |
| 5/1 | 5/2 | ... | ... | ... | ... |
| 6/1 | ... | ... | ... | ... | ... |

这张表格的填写顺序为Z字型填写：1/1, 1/2, 2/1, 3/1, 2/2, 1/3, 1/4, 2/3, ...
请问表格中第n个填写的是几分之几呢？注意不需要约分。

输入样例：

7

输出样例：

1/4

输入样例：

10

输出样例：

4/1

代码1

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main(){
4      int n;
5      cin>>n;
6      bool flag=1;
7      int a=
8      int b=
9      for(int i=2;i<=n;i++){
10         if(flag){
11             if(a==1){
12                 b++;
13                 
14             }else{
15                 b++;
16                 a--;
17             }
18         }
19     }
20     if() {
21         
22         flag=!flag;
23     }else{
24         
25         
26     }
27 }
28
29 cout<<a<<"/"<<b<<endl;
30 return 0;
31 }
```

代码2

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main(){
4      int n;
5      cin>>n;
6      int id=1;
7      int cnt=0;
8      while(cnt+id<n){
9          cnt+=id;
10         
11     }
12     if(id%2==0)
13         cout<<n-cnt<<"/"<<.<<endl;
14     else
15         cout<<<<"/"<<n-cnt<<endl;
16     return 0;
17 }
```

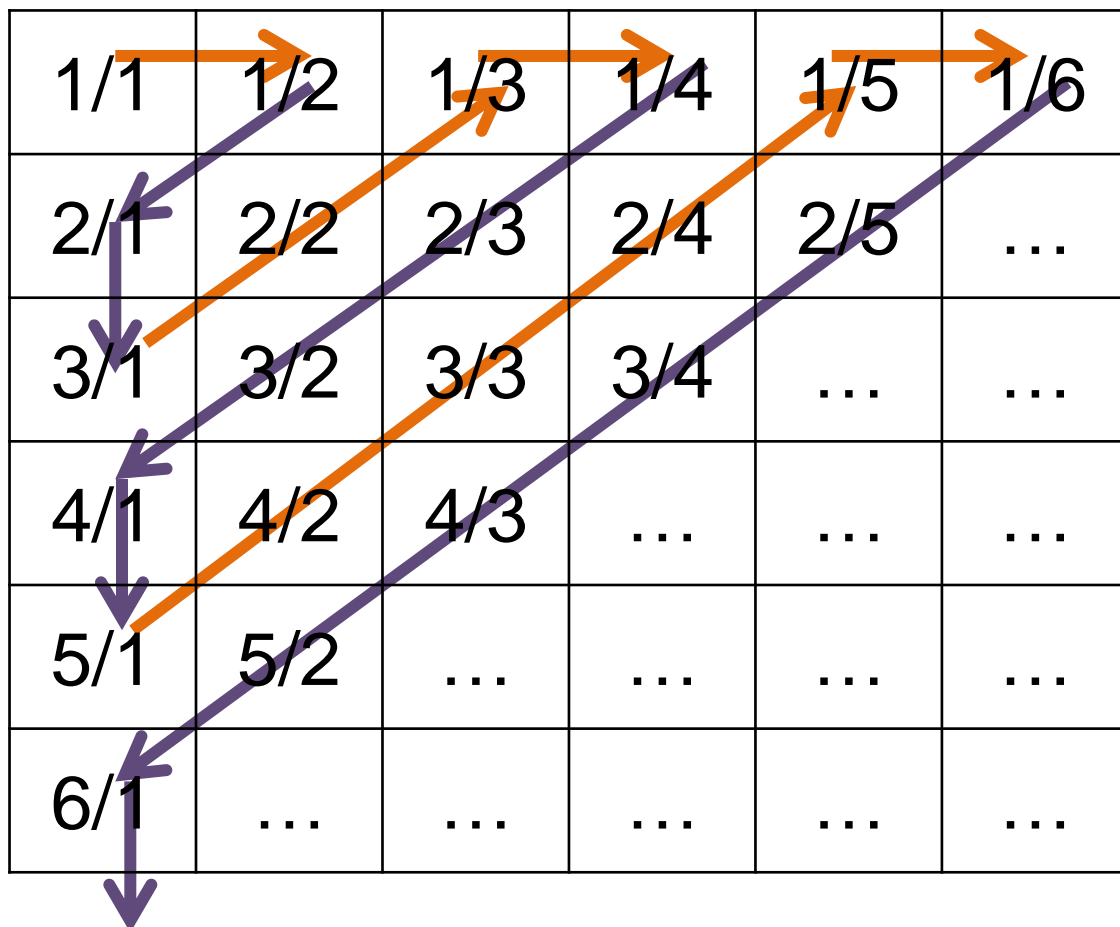
请观察
变量的生命线

观察id一生
你有什么结论

有理数

数学家康托证明了有理数（某整数除以某整数）是可枚举的。他用了下面这一张表来说明任意给定的有理数都会在表格里出现。输入一个正整数 n ， $n \leq 10000000$ 输出一个分数

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 |
| 2/1 | 2/2 | 2/3 | 2/4 | 2/5 | ... |
| 3/1 | 3/2 | 3/3 | 3/4 | ... | ... |
| 4/1 | 4/2 | 4/3 | ... | ... | ... |
| 5/1 | 5/2 | ... | ... | ... | ... |
| 6/1 | ... | ... | ... | ... | ... |



输入样例
7

输出样例
1/4

纯模拟： $O(N)$

根据题意，循环 n 次，每次填一个新数

$n \leq 10000000$ ，可以通过

模拟+简化: $O(\sqrt{N})$

定位第n个数在第几条斜线上?

id号斜线朝向左下还是右上?

1号斜线到(id-1)号斜线共几个数?

该斜线编号为id

由 $id \% 2$ 决定

总数记为cnt

$$= id * (id - 1) / 2$$

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 |
| 2/1 | 2/2 | 2/3 | 2/4 | 2/5 | ... |
| 3/1 | 3/2 | 3/3 | 3/4 | ... | ... |
| 4/1 | 4/2 | 4/3 | ... | ... | ... |
| 5/1 | 5/2 | ... | ... | ... | ... |
| 6/1 | ... | ... | ... | ... | ... |

6号斜线

分母最大是6

分子最大是6

模拟+简化: $O(\sqrt{N})$

定位第n个数在第几条斜线上?

该斜线编号为id

id号斜线朝向左下还是右上?

由id%2决定

1号斜线到(id-1)号斜线共几个数?

总数记为cnt

$= id*(id-1)/2$

```
6   int n,id=1,cnt=0;
7   cin>>n;
8   while(cnt+id<n)
9       cnt+=id,id++;
10  if(id%2==0)
11      cout<<n-cnt<<"/"<<id-(n-cnt)+1<<endl;
12  else
13      cout<<id-(n-cnt)+1<<"/"<<n-cnt<<endl;
14  return 0;
15 }
```

id号斜线朝向左下

id号斜线朝向右上

金币

国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币.....；这种工资发放模式会一直这样延续下去：当连续 N 天每天收到 N 枚金币后，骑士会在之后的连续 $N+1$ 天里，每天收到 $N+1$ 枚金币。请计算在前 K 天里，骑士一共获得了多少金币。 $K \leq 10000$

输入只有1行，包含一个正整数 K ，表示发放金币的天数。输出只有1行，包含一个正整数，即骑士收到的金币数。

| | | |
|------------|-------------|----------------|
| 输入样例： 2 | 输入样例： 5 | 输入样例： 1000 |
| 输出样例： 3 | 输出样例： 11 | 输出样例： 29820 |
| 输入样例： 3 | 输入样例： 6 | |
| 输出样例： 5 | 输出样例： 14 | |



代码1

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int k;
5      int sum=
6      int n=1;
7      int cnt=0;
8      cin>>k;
9      for(int i=1; ; i++){
10         sum+=
11         if( ){
12             cnt=0;
13             n++;
14         }
15     }
16     cout<<sum<<endl;
17     return 0;
18 }
19 }
```

第一步：该干什么？

| | | |
|----|-------|--------------|
| 输入 | k代表 | 一共要发几天金币 |
| 输出 | sum代表 | 已收到总金币数 |
| | n代表 | 当天发几个金币 |
| | cnt代表 | 目前这种金币已经发了几天 |

第二步：尝试翻译每一行写注释

第三步：观察变量们的生命线

| | |
|--------|------------------------------------|
| k的一生 | 输入以后没有用到! |
| sum的一生 | 第10行加上某个金币数 |
| n的一生 | 变化规律n=1,2,3,4,... |
| cnt的一生 | 第7,14行清零 猜测要有变化 哪一个空适合变化cnt? |

代码2

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int k;
5      cin>>k;
6      int day=
7      int n=1;
8      int sum=0;
9      while(day+n<=k){
10         sum+=
11         day+=n;
12         
13     }
14     sum+=
15     cout<<sum<<endl;
16     return 0;
17 }
```

纸和笔写出
day代表什么含义

纸和笔完成
所有填空推理

螺旋填表

手写一个如下图所示螺旋方阵是一件很容易的事，但是计算机打印出来就不容易了，因为打印机的打印头只能前进不能后退.解决的方法是认真观察这个方阵，从中发现规律，不难发现，螺旋方阵是从里到外进行填数，每一层的初始位置都固定在左上角，每一层又可分为下、右、上、左四个部分。运用数组下标变化规律来控制填数，保证数据按螺旋线顺序存入数组。最后打印这个数组。

输入：

5

输出：

```
1 16 15 14 13
2 17 24 23 12
3 18 25 22 11
4 19 20 21 10
5 6 7 8 9
```

```

1  #include<iostream>
2  using namespace std;
3  const int N=109;
4  int n,a[N][N],b[2]={0,1};
5  int main() {
6      cin>>n;
7      int k=n, t=1, p=1;
8      for(int i=1; [ ] i++) {
9          b[p/(k+1)]+=t;
10         a[b[0]][b[1]]=i;
11         [ ]
12         if([ ]) {
13             [ ]
14             k=k-1;
15             [ ]
16         }
17     }
18     for(int i=1;i<=n;i++){
19         for(int j=1;j<=n;j++)
20             cout<<a[i][j]<<" ";
21         cout<<endl;
22     }
23     return 0;
24 }

```


太戈编程

270,766

拓展题1411