

信奥算法

数位分离

```
int x=142857;
```

取出最后1位

$x \% 10$

7

取出最后2位

$x \% 100$

57

取出最后3位

$x \% 1000$

857

取出最后4位

$x \% 10000$

2857

int x=142857;

删除最后1位

x/=10;

14285

删除最后2位

x/=100;

1428

删除最后3位

x/=1000;

142

删除最后4位

x/=10000;

14

```
int x=142857;
```

取出个位数字

$x \% 10$

7

取出十位数字

$x / 10 \% 10$

5

取出百位数字

$x / 100 \% 10$

8

取出千位数字

$x / 1000 \% 10$

2

```
int x=142857;
```

取出个位数字

```
x%10
```

7

取出十位数字

```
x%100/10
```

5

取出百位数字

```
x%1000/100
```

8

取出千位数字

```
x%10000/1000
```

2

```
int x=142857;
```

```
x=x*10;
```

```
x=x*10+1;
```

```
x=x*10+2;
```

```
x=x*10+3;
```

```
cout<<x;
```

预测
输出
结果

求位数

输入一个非负整数 $n(0 \leq n \leq 10^{18})$ ，输出 n 是几位数。

输入样例：
1234567890

输入样例：
0

输入样例：
233

输出样例：
10

输出样例：
1

输出样例：
3

1位数

0,1,2,3,4,5,6,7,8,9

2位数

10,11,12,13,14,15,16,17,18,19,...,98,99

3位数

100,101,102,103,104,105,106,...,998,999


```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll size(ll x){
5      ll ans=0;
6      do{
7          ans++;
8          x/=10;
9      }while(x);
10     return ans;
11 }
12 int main(){
13     ll n;
14     cin>>n;
15     cout<<size(n)<<endl;
16     return 0;
17 }

```

长整数类型
简写为ll

分号;不能漏

数位分离

do while语句
先执行再判断

数字0也算1位

```
1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll size(ll x){
5      ll ans=0;
6      do{
7          ans++;
8          x/=10;
9      }while(x);
10     return ans;
11 }
12 int main(){
13     ll n;
14     cin>>n;
15     cout<<size(n)<<endl;
16     return 0;
17 }
```

ans代表当前位数

删除末尾

删除个位数

```
1 #include<iostream>
2 #include<string>
3 using namespace std;
4 int main(){
5     string s;
6     cin>>s;
7     cout<<s.size()<<endl;
8     return 0;
9 }
```

任何信息都可以
用字符串输入储存

哪怕题目描述为"整数"

错误代码

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

错在哪里

测试哪个数据?

求二进制位数

输入一个非负整数 n ($0 \leq n \leq 10^{18}$)，输出 n 的二进制是几位数。

输入样例：

3

输入样例：

4

输入样例：

8

输出样例：

2

输出样例：

3

输出样例：

4

3的二进制是11

4的二进制是100

8的二进制是1000

```
1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll size(ll x){
5      ll ans=0;
6      do{
7          ans++;
8          x/=2;
9      }while(x);
10     return ans;
11 }
12 int main(){
13     ll n;
14     cin>>n;
15     cout<<size(n)<<endl;
16     return 0;
17 }
```

删除二进制末尾

有多少0

输入一个非负整数 n ($0 \leq n \leq 10^{18}$)，输出 n 的数字包含几个0。

输入样例：

202

输出样例：

1

输入样例：

0

输出样例：

1

输入样例：

500

输出样例：

2

错误程序

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      long long n,cnt=0;
5      cin>>n;
6      while(n>0){
7          if(n%10==0)cnt++;
8          n/=10;
9      }
10     cout<<cnt<<endl;
11     return 0;
12 }
```

错在哪里


```
1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll count0(ll x){
5
6
7
8
9
10
11 }
12 int main(){
13     ll n;
14     cin>>n;
15     cout<<count0(n)<<endl;
16     return 0;
17 }
```

该函数有什么功能?

count0返回值代表
x里有几个0

共有多少0

输入一个非负整数 n ($0 \leq n \leq 10^{18}$)，输出0到 n 共包含几个0。

输入样例：

0

输出样例：

1

输入样例：

11

输出样例：

2

输入样例：

100

输出样例：

12

0

10

0,10,20,30,40

50,60,70,80,90

100

```

1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll count0(ll x){
5
6
7
8
9
10
11 }
12 int main(){
13     ll n,sum=0;
14     cin>>n;
15     for(int i=0;i<=n;i++)
16         sum+=count0(i);
17     cout<<sum<<endl;
18     return 0;
19 }

```

请完成5-10行
老师检查

请翻译

几个数包含0

输入一个非负整数 n ($0 \leq n \leq 10^{18}$), 输出0到 n 共有几个数包含0。

输入样例:

0

输出样例:

1

输入样例:

100

输出样例:

11

输入样例:

110

输出样例:

21

如何利用
`count0()`函数

几个数不包含1

输入一个非负整数 $n(0 \leq n \leq 10^{18})$,输出0到 n 共有几个数不包含1。

输入样例:

10

输出样例:

9

输入样例:

20

输出样例:

10

输入样例:

22

输出样例:

11

参考上一题
`count0()`函数

如何设计
`count1()`函数

```
1  #include<iostream>
2  using namespace std;
3  typedef long long ll;
4  ll count1(ll x){
5      ll cnt=0;
6      do{
7          if(x%10==1)cnt++;
8          x/=10;
9      }while(x);
10     return cnt;
11 }
12 int main(){
13     ll n,ans=0;
14     cin>>n;
15     for(int i=0;i<=n;i++)
16         if(count1(i)==0) ans++;
17     cout<<ans<<endl;
18     return 0;
19 }
```

该函数有什么功能?

count1返回值代表
x里有几个1

请翻译

233

输入一个非负整数 n ($0 \leq n \leq 10^{18}$), 判断其中是否有一个2和两个3.

输入样例:

33321

输出样例:

Yes

输入样例:

222223

输出样例:

No

输入样例:

233

输出样例:

Yes

如何设计函数
完成数位分离?

该函数有什么功能?

```
4  ll count(ll x, ll k){  
5  
6  
7  
8  
9  
10  
11 }  
12 int main(){  
13     ll n;  
14     cin >> n;  
15     if(count(n, 2) >= 1 && count(n, 3) >= 2)  
16         cout << "Yes" << endl;  
17     else  
18         cout << "No" << endl;  
19     return 0;  
20 }
```

补全
程序

数字反转

给定一个整数 n ，请将该数各个位上数字反转得到一个新数。
新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零。注意负号要保留。

数据范围： $-1,000,000,000 \leq n \leq 1,000,000,000$

输入样例：

123

输出样例：

321

输入样例：

110

输出样例：

11

输入样例：

-380

输出样例：

-83

难点1

负号的处理

难点2

末尾0的处理

数字反转

```
11 n,ans=0;
cin>>n;
if(n<0){
    cout<<"-";
    n=-n;
}
do{
    ans=ans*10+n%10;
    n/=10;
}while(n);
cout<<ans<<endl;
```

取出n的末位
添加到ans尾部

删除n末位

公里数

某辆车的里程表出现了故障：它总是跳过数字3和数字8。也就是说，当前显示已走过两公里时，如果车子再向前走一公里，那么将显示4公里，而不是3公里（数字3跳过了）。再比如，当前是15229公里，车子再向前走一公里，显示的是15240公里，而不是15230公里。数字8也同样跳过。

输入故障里程表显示的公里数 s （ $1 \leq s \leq 1000000$ ），输出正确的公里数。

输入样例：

15

输出样例：

12

输入样例：

2

输出样例：

2

1到15数字中
包含数字3、8、13



限速5公里

该函数有什么功能?

```
4 ll countk(ll x, ll k){  
5       
6     do{  
7         if(x%10==k) cnt++;  
8         x/=10;  
9     }while(x);  
10    return   
11 }  
12 int main() {  
13     ll s, ans=0;  
14     cin>>s;  
15     for(ll i=1; i<=s; i++)  
16         if()  
17             ans++;  
18     cout<<ans<<endl;  
19     return 0;  
20 }
```

补全
程序

幸运的58

小明对数字“5”和“8”非常感兴趣，认为是幸运数字，那么仅用幸运数字组成的整数就是幸运整数，请你帮助小明找出从5开始从小到大第 k 个幸运整数。

输入仅一行包含一个正整数 k ($k \leq 300$)。
输出仅一行包含一个正整数，为第 k 个幸运整数。

用概括题目大意
写出已知什么求什么
不能遗漏核心要点

要求：可以只看概况缩写后的
题目大意，理解并做题

已知正整数 k ， $k \leq 300$
求只包含5和8的正整数里第 k 小数

如何设计函数
完成数位分离？

幸运的58

```
13 int main() {  
14     ll k,n=5,i=1;  
15     cin>>k;  
16     while(i<k){  
17         n++;  
18         if(lucky(n))  
19             i++;  
20     }  
21     cout<<n<<endl;  
22     return 0;  
23 }
```

i代表当前有几个幸运整数

n代表当前第i个幸运整数是几

当目前幸运整数不够k个

如果n是幸运整数

幸运的58

4
5
6
7
8
9
10
11
12

```
bool lucky(int x){  
    do{  
        int r=x%10;  
        if(  
            return 0;  
        x/=10;  
    }while(x);  
    return 1;  
}
```

判断x里是否全是58

取出x的个位

补全程序

删除x的个位

作业要求

0.翻译每一行写注释

1.每次出现错误都必须在微信班级群发错误笔记，含姓名，题号，标题，错误原因，如何避免

2.将该笔记记录在自己电脑的文档里，作为经验累积

举例：

姓名：黄晓明

题号：12

标题：长方形体积

错误原因：类型选择错误，应该用double浮点数

避免方法：每次定义时思考数据类型，提交前测试多组自编输入数据检验

太戈编程

216

790

797

拓展题

217, 218, 132, 277