

### 1、算式

遍历字符，遇到“+”号就前后相加，遇到“-”号就前后相减。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string s;
    char a;
    int ans=0,x=0,i=0;
    cin>>s;
    while(i<s.size()){
        if(!isdigit(s[i])) a=s[i],i++;
        while(isdigit(s[i]) && i<s.size()){
            x=10*x+s[i]-48;
            i++;
        }
        if(a=='+') ans+=x; else if(a=='-') ans-=x; else ans=x;
        x=0;
    }
    cout<<ans<<endl;
    return 0;
}
```

### 2、可乐

可以模拟，也可以数学分析。

因为 k 个可乐瓶=1 瓶可乐+1 个可乐瓶，所以 k-1 个可乐瓶=1 瓶可乐。即  $n+(n-1)/(k-1)$ 。

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n,k;
    cin>>n>>k;
    cout<<n+(n-1)/(k-1);
    return 0;
}
```

### 3、下棋

如果 n 为偶数则 n 的平方也为偶数，奇数的平方也为奇数。

当 n 为偶数时我们可以把整个棋盘分成多个 1\*2 的方形，因为都是以最优解来走的，所以后手每次都会走到一个新的方形里，给先手留下一个空位，所以后手必败。

当 n 为奇数时我们可以把除了起点外整个棋盘分成多个 1\*2 的方形，那么每次就是先手走到新的方形里，所以先手必败。

综上所述我么只需要每次判断 n 的奇偶性就行了。

```
#include < bits/stdc++.h >
int n;
int main()
{
```

```

while (1) {
    cin>>n;
    if (n==0) break;
    if (n%2) cout<<"zhi"<<endl;
    else cout<<"neng"<<endl;
}
return 0;
}

```

#### 4、真人秀

根据题目要求，要查找的是符合要求的末尾匹配字符串。因为是找最小串，所以要先排序，再遍历查找，找到的第一个字符串即为答案。

```

#include<bits/stdc++.h>
using namespace std;
string s[1010];
bool cmp(const string s1,const string s2) { //不符合从小到大返回 false
    int t1=s1.length();
    int t2=s2.length();
    if(t1>t2)return false;
    else if(t1<t2)return true;
    else {
        int i;
        for(i=0; i<t1; i++) {
            if(s1[i]>s2[i])return false;
            else if(s1[i]<s2[i])return true;
        }
        return true;
    }
}
int main() {
    int n,q,i,j,x,len;
    cin>>n>>q;
    string str1;
    for(i=1; i<=n; i++) {
        cin>>s[i];
    }
    sort(s+1,s+n+1,cmp);
    for(i=1; i<=q; i++) {
        cin>>x>>str1;
        for(j=1; j<=n; j++) {
            len=s[j].rfind(str1);
            if(len!=-1&&len+x-1==s[j].length()-1) {
                cout<<s[j]<<endl;
                break;
            } else if(j==n)cout<<-1<<endl;
        }
    }
}

```

```

    }
}
return 0;
}

```

## 5、飞船

尽量将燃料瓶使用时间平均分到两边，先从小到大排序，所有燃料时间和  $s$ 。

正常情况：两边平均分配，答案为  $s/2$ 。

特殊情况：最后一瓶大于前面所有燃料时间总和，答案为  $s-a_n$ 。

```

#include <bits/stdc++.h>
using namespace std;
int a[1005];
int main()
{
    int n;
    while(cin >> n)
    {
        int s=0;
        for (int i=1;i<=n;i++)
        {
            cin >> a[i];
            s+=a[i];
        }
        sort(a+1,a+n+1);
        double ans = s-a[n];
        if (ans > a[n]) ans = s/2.0;
        cout << fixed << setprecision(1) << ans << endl;
    }
    return 0;
}

```

## 6、翻滚的长方体

由于计算的是最少操作次数，尽量让‘长’着地，那么：

(1)一次放到+一次立起的长度=3

(2)一次放到+一次立起的次数=2

(3)看是否能被 3 整除，若不能那就用‘宽’着地进行翻滚。则得到：

$ans=((n-1)/3+(m-1)/3+(n-1)\%3+(m-1)\%3)\times 2$

特殊情况:翻长方体不能越界，由于  $n\leq m$ ，所以当  $n=1$  或  $n=2$  时，若  $(m-1)\%3\neq 0$ ，则输出 impossible。

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int N,M;
    cin>>N>>M;
    if(N>M) swap(N,M);
}

```

```

if(N==1)
{
    if((M-1)%3) puts("impossible");
    else cout<<(M-1)/3*2<<endl;
    return 0;
}
if(N==2)
{
    if((M-1)%3)
        puts("impossible");
    else
        cout<<(M-1)/3*2+1<<endl;
    return 0;
}
if(N==3)
{
    if(M==3)
        puts("8");
    else if((M-1)%3)
        cout<<(M-1)/3*2+2+2+(M-1)%3<<endl;
    else
        cout<<(M-1)/3*2+2<<endl;
    return 0;
}
cout<<(M-1)/3*2+(M-1)%3+(N-1)/3*2+(N-1)%3<<endl;
return 0;
}

```