

pA:麋駘薨

# 題目大意

優格製作包含了 $n/10$ 的lp33和 $9n/10$ 的鮮奶

發酵lp33和鮮奶的時間不同

給你一個 $n$ 和一個時間，問能不能在時間內做出優格

這題是考驗各位基本的小數運算和簡單的條件判斷  
=>水題

# 陷阱

很多人看到題目中說到會輸入一個整數n  
但是很明顯，當我們將 $n*9/10$ 大多時候會是小數  
而即使用double儲存，也會因為是用整數計算會出問題

例如：

`double d=9*3/10`

會因為先計算右側得到 $27/10=2$

最後d也只會拿到2

=>把n也用double輸入就沒問題啦!

# AC CODE

```
int main() {  
    double n,m;  
    cin>>n>>m;  
    double Time=(n/10)*3+(9*n/10)*2;  
    if(Time<=m){  
        cout<<"Loli Loli!\n";  
    }  
    else if(Time-m<5){  
        cout<<"Q_3lpp13_Q\n";  
    }  
    else{  
        cout<<"Hentai!\n";  
    }  
    return 0;  
}
```

**PB: 盲目的流行**

# 主要考點

- 二維陣列

基本ㄉ東西

- 迴圈

對陣列輸入輸出

- 數學:(~~廣義角~~)

廣義角的部分刪掉了 所以角度只有

$\text{sit}\alpha = 0, 90, 180, 270, 360$

# AC code

輸入:

二維陣列的輸入是基本了  
數字用int即可

```
1  #include<iostream>
2  using namespace std;
3  int h[3000][3000];
4  int main() {
5
6      int n,k,sita;
7      cin>>n>>k>>sita;
8      for(int m=0;m<n;m++){
9          for(int m2=0;m2<k;m2++){
10             cin>>h[m][m2];
11         }
12     }
13     sita=sita/90;
14     sita=sita%4;
15     if(sita==0){
16         for(int m=0;m<n;m++){
17             for(int m2=0;m2<k;m2++){
18                 if(m2!=0){
19                     cout<<" ";
20                 }
21                 cout<<h[m][m2];
22             }
23             cout<<"\n";
24         }
25     }
```



# 計算輸出

輸出部分:

聰明的人應該有發現

這題不需要運算

只要換個角度(方向)輸出即可

一樣是二維陣列的練習

要注意的是 要小心sita角的部分超出360度

直接四個if else即可

switch case也可

(0,0)	...	(0,m2-1)
...		...
(m-1,0)	...	(m-1,m2-1)

```
13 sita=sita/90;
14 sita=sita%4;
15 if(sita==0){
16     for(int m=0;m<n;m++){
17         for(int m2=0;m2<k;m2++){
18             if(m2!=0){
19                 cout<<" ";
20             }
21             cout<<h[m][m2];
22         }
23         cout<<"\n";
24     }
25 }else if(sita==1){
26     for(int m2=0;m2<k;m2++){
27         for(int m=n-1;m>=0;m--){
28             if(m!=n-1){
29                 cout<<" ";
30             }
31             cout<<h[m][m2];
32         }
33         cout<<"\n";
34     }
35 }else if(sita==2){
36     for(int m=n-1;m>=0;m--){
37         for(int m2=k-1;m2>=0;m2--){
38             if(m2!=k-1){
39                 cout<<" ";
40             }
41             cout<<h[m][m2];
42         }
43         cout<<"\n";
44     }
45 }else if(sita==3){
46     for(int m2=k-1;m2>=0;m2--){
47         for(int m=0;m<n;m++){
48             if(m!=0){
49                 cout<<" ";
50             }
51             cout<<h[m][m2];
52         }
53         cout<<"\n";
54     }
55 }
56 return 0;
57 }
```



PC: 嘿嘿

## subtask1 n=1

你發現了嗎

輸出一坨星星就有11分欸

好棒喔

```
for(i=0;i<n*k;i++)  
{  
    for(j=0;j<n*k;j++)  
    {  
        cout<<'*';  
    }  
    cout<<'\\n';  
}
```

## subtask3 k=1

你發現了嗎

只要看 $i+j$ 是不是偶數就行了欸

```
for(i=0;i<n*k;i++)
{
    for(j=0;j<n*k;j++)
    {
        if((i+j)%2==0)
            cout<<'*';
        else
            cout<<' ';
    }
    cout<<'\n';
}
```

```
  01234
0* * *
1 * *
2* * *
3 * *
4* * *
```

# subtask5 ouo

你發現了嗎

只要看 $i/k + j/k$ 是不是偶數就行了欸

```
for(i=0;i<n*k;i++)
{
    for(j=0;j<n*k;j++)
    {
        if((i/k+j/k)%2==0)
            cout<<'*';
        else
            cout<<' ';
    }
    cout<<'\n';
}
```

	0	1	2	3	4	5
0	**				**	
1	**				**	
2			**			
3			**			
4	**				**	
5	**				**	

# PD: 留社考題的保密措施

- 沒錯，這是梗題

# 先來看看哪些東西是在唬爛的

- 看看第三步，這個數字是不是大的有點誇張
  - 先假設兩個正整數A、B，而A小於B，則 $A \% B$ 必為A
  - EX:  $21 \% 22 = 21$
  - 因此，第四步可忽略
  - 所以，第三步也不用算
  - 故第一步其實也不需要解
- 
- **結論：只要算第二步就行了**



# 所以.....要怎麼處理第一個正整數

- 就算用不到第一個輸入的數字，第一個數字還是會輸入，而且不小，可能會超出int
- 直接把要接收第一個輸入的變數定義為string來解決

• **結論：用string**

# 那麼第二步要怎麼解

- 來源：2019年TRML接力賽第二回合第三棒（換句話說這題三分鐘內就可以解出來）
- 差異：原題是正 $n$ 邊形，PD是正 $2n$ 邊形（ $n$ 為正整數且保證圖形是畫得出來的，例如不會有正2邊形）
- 若為正 $2n+1$ 邊形，則畫不出直角三角形（後面會討論），因此為了降低本題梗度，才直接用正 $2n$ 邊形

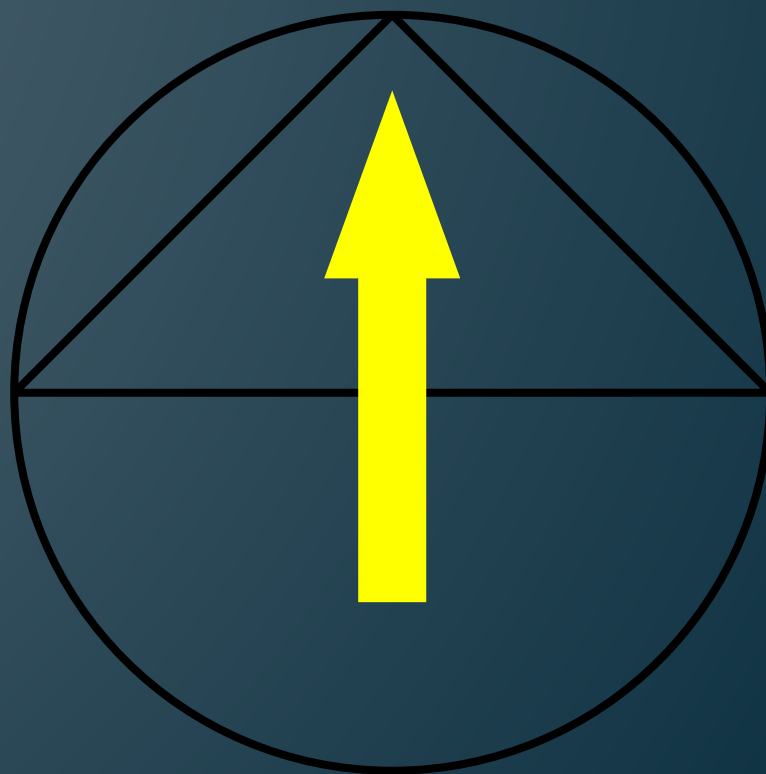
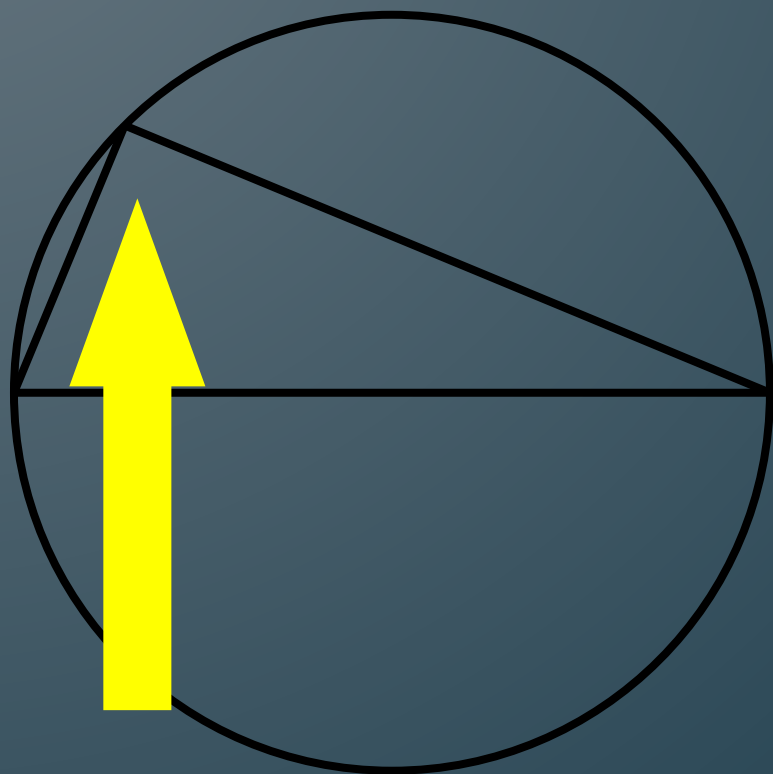
# 那麼第二步要怎麼解：教科書解法

- 正 $n$ 邊形的 $n$ 個點必共圓
- 因此，對正 $2n$ 邊形，可畫的直角三角形數量有( $4$ 乘以 $C$ 的 $n$ 取 $2$ )個
- 正 $2n+1$ 邊形則畫不出直角三角形，因為找不到圓周角其對應的弧度為 $180$ 度

# 那麼第二步要怎麼解：數學菜雞的解法

- ~~沒錯，我就是那個數學菜雞~~
- 正 $n$ 邊形必為圓內接正 $n$ 邊形
- 只要在上面找到兩點，連起來剛好是外接圓的直徑，再加上任意一個點就是直角三角形

● **△國中數學警告△**



# 那麼第二步要怎麼解：數學菜雞的解法

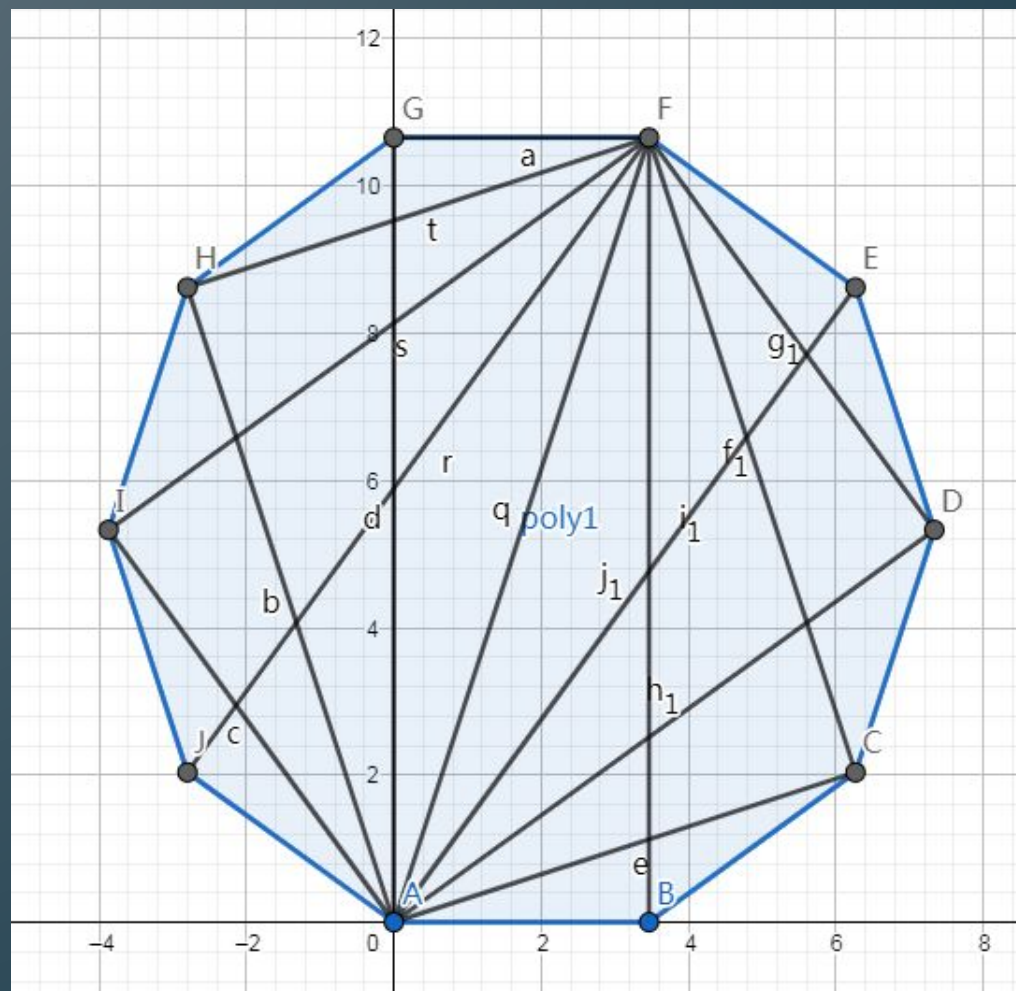
- 對正 $2n+1$ 邊形而言，在上面找不到兩點，連起來剛好是外接圓的直徑
- 換句話說，沒有任何一條外接圓的直徑會同時通過正 $2n+1$ 邊形的兩個點
- 所以正 $2n+1$ 邊形畫不出直角三角形



# 那麼第二步要怎麼解：數學菜雞的解法

- 而正 $2n$ 邊形卻有 $n$ 條相異直徑是同時通過兩個點，為直角三角形中，直角的對邊，所以不同條直徑不會做出相同直角三角形
- 因此，只要再挑選除了剛剛被用掉作直徑的兩點以外的任意一點（共 $2n-2$ 個點）就能作直角三角形
- 故共有 $(n(2n-2))$ 個直角三角形

# 那麼第二步要怎麼解：數學菜雞的解法



# 那麼第二步要怎麼解：數學菜雞的解法

- 加上這題不用考慮正 $2n+1$ 邊形
- 所以不用判斷奇偶數，直接用 $n(2n-2)$
- 結論：對於第二個輸入的正整數 $a$ ，輸出 $a(2a-2)$

# 可是我只有60分

- $n(2n-2)$ 真的不是小數字
- 例如 $n$ 為32769時,  $n(2n-2)=21\,475\,491\,84$
- $21\,475\,491\,84 > 21\,474\,836\,47$
- 所以接收第二個正整數的變數用int的話會爆掉
- 所以你可以用long long或long long int, 這兩個的上限是一樣的
- 最好是直接用unsigned long long int, 這樣的上限大約是前者的兩倍

**為什麼？為什麼極靈你不幫我發大絕？**

# 為什麼？為什麼我拿不到AC？

- 因為最後面的測資範圍已經超出`unsigned long long int`很多了
- 以最後一項子任務為例，五筆測資分別為
- 1791095846 4282876141234120
- 3093770125 40053033691234
- 491264 55029031434565
- 1298508492 4290846342767856
- 630311760 1013994434532453
- 以最後一筆為例，答案為  
2056369426525576198044551329512
- 因此，經過 $n(2n-2)$ 的運算後就直接爆炸了

# 如何拿AC:大數處理

- 當數字太大的時候, 就不能直接運算, 不然會爆掉



# ● △國小數學警告△

# 簡單的四則運算教學

- 還記得加減法和乘法的直式運算嗎？
- 簡單來說，就是把直式運算的流程程式化
- 因為 $n(2n-2)$ 小於 $n(2n)$ ，即 $2n^2$ ，所以一個 $k$ 位數的正整數下去做 $n(2n-2)$ 的運算結果不會大於 $2k+1$ 位數
- 建立兩個長度為 $k$ 的陣列（一個接收輸入，一個用來存減1後的數）、一個長度 $2k+1$ 的陣列，每一格存一個小於10的正整數
- 利用直式運算的原理把運算結果存到長度 $2k+1$ 的陣列

# 簡單的四則運算教學

Maths

$$24 \times 7 = 168$$

	百位	十位	個位
		2	4
×			7
<hr/>			
		2	8
	1	4	0
<hr/>			
	1	6	8

個位：4 (個一)  $\times 7 = 28$  (個一)

十位：2 (個十)  $\times 7 = 14$  (個十)

# 簡單的四則運算教學

	陣列一		2	4
X	陣列二		3	7
			$7 \times 2 = 14$	$7 \times 4 = 28$
		$3 \times 2 = 6$	$3 \times 4 = 12$	
	陣列三	6	$14 + 12 = 26$	28
處理進位	陣列三	6	$26 + 28 / 10$	$28 \% 10$
		$6 + 28 / 10$	$28 \% 10$	8
		8	8	8

# 簡單的四則運算教學：程式化

```
for(int i=n-1;i>=0;i--)  
{  
    for(int j=n-1;j>=0;j--)  
    {  
        v2[j+i+2]+=v1[j]*v[i];  
    }  
    for(int i=2*n;i>0;i--)  
    {  
        v2[i-1]+=v2[i]/10;  
        v2[i]=v2[i]%10;  
    }  
}
```

- 拜託減法記得退位

# 減法退位：向上一個位數借10

```
for(int i=n-1;i>0;i--)  
{  
    if(v1[i]<0)  
    {  
        v1[i-1]--;  
        v1[i]+=10;  
    }  
}
```



# 輸出時的注意事項

- 可能會有空的位置: 在建立陣列時通通塞0, 最後輸出時忽視前面的0

```
bool b=0;
for(int i=0;i<n;i++)
{
    if(b==0&&a[i]=0)
    {
        continue;
    }
    else
    {
        b=1;
        cout<<a[i];
    }
}
```

# 回到源頭：如何輸入

```
string a;  
int n;  
cin>>a>>a;  
n=a.length();  
int v[n],v1[n],v2[2*n+1];  
for(int i=0;i<n;i++)  
{  
    v[i]=a[i]-'0';  
    v1[i]=a[i]-'0';  
}
```

# 回到源頭：如何輸入（使用STL容器）

```
deque<int>v,v1,v2;  
    string a;  
    int n=0;  
    cin>>a>>a;  
    while(a[n]!='\0')  
    {  
        v.push_back((int)a[n]-48);  
        v1.push_back((int)a[n]-48);  
        n++;  
    }
```

# AC程式總覽

```
#include<iostream>
using namespace std;
int main(){
    string a;
    int n;
    cin>>a>>a;
    n=a.length();
    int v[n],v1[n],v2[2*n+1];
    for(int i=0;i<n;i++)
    {
        v[i]=a[i]-'0';
        v1[i]=a[i]-'0';
    }
    v1[n-1]--;
    for(int i=n-1;i>0;i--)
    {
        if(v1[i<0])
        {
            v1[i-1]--;
            v1[i]+=10;
        }
    }
}
```

# AC程式總覽

```
23     for(int i=0;i<2*n+1;i++)
24     {
25         v2[i]=0;
26     }
27     for(int i=n-1;i>=0;i--)
28     {
29         for(int j=n-1;j>=0;j--)
30         {
31             v2[j+i+2]+=2*v1[j]*v[i];
32         }
33         for(int i=2*n;i>0;i--)
34         {
35             v2[i-1]+=v2[i]/10;
36             v2[i]=v2[i]%10;
37         }
38     }
39     n=2*n+1;
40     int t=0;
41     for(t;v2[t]==0;t++){ } //測出第幾位開始不為0
```

# AC程式總覽

```
39     n=2*n+1;
40     int t=0;
41     for(t;v2[t]==0;t++){ } //測出第幾位開始不為0
42     for(t;t<n;t++)
43     {
44         cout<<v2[t];
45     }
46     cout<<endl;
47
48
49
50     return 0;
51 }
```

這樣就能AC了

pE: ub33買lp33



# PE ub33買lp33

題意說明:對於給定 $l, q$  找到最小的 $r$  使得區間和  $\geq q$

照著題目要求做試試 從 $l$ 開始一個一個加 超過就停

複雜度  $O(NQ)$   $N, Q \leq 300000$  這樣會TLE

試著二分搜? 以前綴和處理後 二分搜右界

對於 $l$ 非1之詢問 減去 $1 \sim l-1$  的前綴和即可

**PF: 開始分裂**

# 題目

- 1.將社部視為 $N \times M$ 格的長方形左下角為 $(1,1)$ 右上角為 $(N,M)$
- 2.有 $P$ 隻母體
- 3.每一次分裂，母體和分裂體會同時朝上下左右沒有母體和分裂體的一格分裂出分裂體
- 4.一格中如有一種母體和分裂體有兩隻以上，會合併成一隻
- 5.一格可塞多種母體和分裂體
- 6.不會往格子外分裂
- 7.一格中的母體不超過 $P / (N \times M) + 10$ 隻

# 解題要點

**1.BFS**

**2.同步BFS**

**3.同步更新**

子題1 ( $N=1, M=1$ )

輸出P個1

## 子題2(P=1)

輸出 $N \times M$

## 子題3(N=1)

(未經驗證)

先對Y座標排序

除了最小和最大外

其值為左邊和自己的距離除以2

加

右邊和自己的距離除以2

## 子題4,5

**BFS**

**建兩個地圖**

**一個紀錄有分裂體的時間**

**一個紀錄同一時間，分裂體有沒有重複  
分裂**

**然後邊BFS邊加起來**

**結束**