

Reference & Hints

一、注意事項

1. 記得開 long long
2. 萬用標頭檔 <bits/stdc++.h>
3. 子題仔細看其實很簡單

二、getline 的用法

一般使用 cin 去做讀取時，會將輸入的資料依照空格的出現拆分成多筆資料。

```
123456 789 11111
```

如果使用 cin 的話，他就會有三個數字，要 cin 三次，值分別為 123456、789、11111。

但若一次讀取整行得到「123456 789 11111」的話，就可以使用 getline，getline 讀取資料會一次讀取一整行(包含空格)，得到的結果會是一個字串，使用方法如下：

```
6 int main() {
7     string s;
8     getline(cin, s);
9 }
```

當然，如果輸入要好幾行你可以 getline 很多次。

需要特別注意的是，同一之程式中要盡量避免同時使用 getline 和 cin，因為這樣會導致 getline 讀取到換行符號而非你要的資料。

假設輸入資料為

```
7
AB CDE FG
```

那如果你先用 cin 讀取 7 再用 getline 讀取第二行，你會發現第二行根本沒被讀取到。

```
7 int n;
8 cin >> n; // 讀取7
9 string s;
10 getline(cin,s); // 讀取第一行7後面的換行符號'\n'
```

如何避免呢？全部都使用 getline 就好或者可以使用 cin.ignore()，不過比較推薦前者做法。

只使用 getline		使用 cin.ignore()忽略第一行的換行符號	
7	string n,s;	7	int n;
8	getline(cin,n);	8	cin >> n;
9	// 注意:讀到的 n 是一個string	9	cin.ignore();
10	int N = stoi(n);	10	string s;
11	// stoi可以把string變成int	11	getline(cin,s);
12	getline(cin,s);		

使用 cin.ignore()有機會因為電腦系統問題導致其他問題，不推薦使用。

三、sort 的用法

非常重要的 sort 用法，在很多題都會用喔！

sort 函式可以把一筆資料由小排到大，舉例來說： $sort([5,2,3,1,4]) = [1,2,3,4,5]$ 。

sort 函式有三個參數 $sort(begin, end, cmp)$ ，其中 $begin$ 為起始位置， end 為終點位置， cmp 為比較函數(compare function)，簡單來說就是把 $[begin, end)$ 的區間依照 cmp 的方式排列。

如果沒有 cmp 就預設是由小排到大。特別注意上面寫的是左閉右開區間，並不包含 end 的範圍，簡單來說就是填你要的範圍的後面一格。

一般 array 的用法(由小排到大)

```
7 int arr[] = {5,2,3,1,4};
8 // arr就代表陣列的第1個位置
9 // arr+5就代表 arr的第1+5個位置
10 // 注意: 沒有包含end的那個範圍
11 sort(arr, arr+5);
```

Vector 的用法(由小排到大)

```
7 vector<int> arr = {5,2,3,1,4};
8 // arr.begin() 就代表陣列的第1個位置
9 // arr.end() 就代表整個 vector最後面再往後一個
10 // 注意: 沒有包含end的那個範圍
11 sort(arr.begin(), arr.end());
```

四、小數點的處理

小數的型態有 $float$ 和 $double$ 兩種，不過不用思考，一律使用 $double$ ，因為它叫做雙倍精度浮點數，數值比較精確，而當題目要求和標準答案不超過 10^{-6} 時，你只要答案盡量準確不要跟正確答案差太多就可以通過，那要怎麼保證盡可能的精確呢？請參考下圖的方式，如果你怕還是不夠準，也可以精確至小數點後更多位。

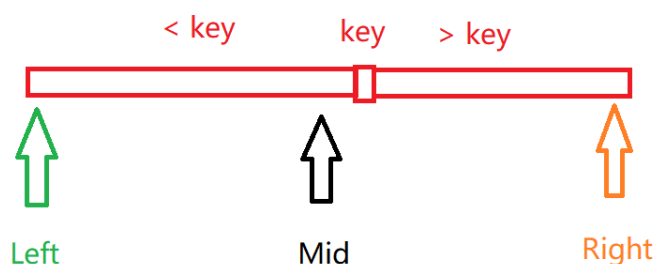
```
7 double ans = 3143.1415926535;
8 cout << fixed << setprecision(6) << ans;
9 // 輸出至小數點後 6 位精確數字(3143.141592)
```

五、關於子題給的測資範圍限制

你可能不知道子題給的比較小的範圍跟原本的有什麼差別。答案就是數字比較小的時候你可以用暴力解。電腦一秒的執行次數是大約 3×10^8 次，而題目的時間限制是 1 秒，因此你要盡可能用有效率的方式去解題，比如說要算 $2^{10000000000}$ ，就會需要 10000000000 次 (10^{10}) 操作，1 秒就跑不完，但是當題目限制次方不會超過 10^6 時，你就可以直接暴力算出要的答案。也就是說當你看到子題有比較小的範圍時，就可以試試看暴力解。或者那個子題就是題目本身很簡單的特例，只需要簡單的處理就能達到分數，比如說：把十進位數字 X 轉乘 Y 進制，但是你可能不會做，但是有一個子題限制 Y 必定是 10，那麼他就變得很簡單。

六、對函數二分搜

二分搜尋(binary search)的概念跟終極密碼的玩法一樣，每次都猜中間，一直到猜到為止。



<https://blog.csdn.net/blowfire123>

說者很容易但是要寫成程式也不簡單，假設要求一個最大的正整數 x_0 使得 $f(x_0) \leq k$ 要怎麼做？其中保證 $f(x)$ 為嚴格遞增函數。

可以簡單地觀察到，因為 $f(x)$ 會隨著 x 值越來越大而變大，因此我們可以找到一個臨界值 x' 使得所有比 x' 小的數，函數值都小於等於 k ，反之，大於 x' 的數函數值都大於 k 。這個 x' 不就是所求的 x_0 嗎？因此我們只要設定左右邊界，然後開始猜中間的數，直到找到臨界值為止。舉例來說： $f(x) = x^2 + 2x + 2$ ，我們要找最大的正整數 x_0 使得 $f(x_0) \leq 18$ ，我們就先設定左界是0，右界是5(因為 $5^2 > 18$)，那我們先猜中間， $f(2) = 4 + 4 + 2 = 10$ ，發現太小，因此把左界設定為2，繼續猜中間，假設猜4，發現 $f(4) = 26$ ，太大，所以已把右邊邊界設成4，繼續猜，2和4中間是3， $f(3) = 9 + 6 + 2 = 17$ ，小於18，因此答案就是3。

```

7 // f(x) = x * x + 2 * x + 2
8 int left = 0, right = 5, target = 18;
9 int ans;
10 // left <= right的意思就是猜到剩一個數字(left = right)
11 while(left <= right) {
12     // mid 就是猜的數字，相加除以2
13     int mid = (left + right)/2;
14     int f_x = mid * mid + 2 * mid + 2;
15     if(f_x <= target) { // 如果太小，就移動右邊界
16         left = mid+1;
17         ans = mid;
18     }
19     else { // 如果太大，就移動左邊界
20         right = mid-1;
21     }
22 }
23 cout << ans;

```


迴圈版

```

9  int mod = 1e9+7; // 就是 10^9+7 = 1000000007
10 int a,n,ans = 1;
11 // if(n == 0) cout<<...;
12 while(n > 0) {
13     // if(n % 2 == 1) ans = (...) % mod;
14     // a = a * a;
15     // n = ...;
16 }

```

遞迴版

```

5  int power(int a,int n) {
6      // if(n == 0) return ...;
7      // if(n % 2 == 1) return (...) % mod;
8      // else {
9          //     int half = ...;
10         //     return (...) % mod;
11     // }
12 }

```

如果不是整數的快速冪整麼辦($a + b\sqrt{c}$)，很簡單，原理一樣，只是變成不同的乘法而已，你只要寫一個 struct 之後再定義他的乘法方式就好。

```

4  struct number {
5      // 定義 number 為 x + (y根號 2)
6      int x,y;
7  }; // 記得要分號
8
9  // 定義乘法
10 number times(number a, number b) {
11     // (x1 + y1根號 2) x (x2 + y2根號 2)
12     // = (x1 x2 + 2 y1 y2) + (x1 y2 + x2 y1)根號 2
13     number result;
14     // result.x = (...) % mod;
15     // result.y = (...) % mod;
16     return result;
17 }

```

快速冪只要把乘法跟類別改掉就好，就是原本是整數的乘法，現在變成你自定義的 struct 乘法去做快速冪。

```

22 number one = {1,0}; // x = 1, y = 0
23 number power(number a, int n) {
24     if(n == 0) return one;
25     // if(n % 2 == 1) return times(...,...);
26     // int half = power(a,n/2);
27     // return tims(...,...);
28 }

```

九、提示

A.崩壞的代理商與無法取得的特點(Unattainable)

1. 二次函數的係數全部都是正的代表他是一個嚴格遞增函數
2. $a = b = 0$ 的子題代表 $f(T) = T^2$
3. A 影院的觀眾進場順序應該很明顯

B.現在正是復權的時刻(DianShiIng)

1. 看著還挺像二微陣列
2. 可以先想想看每個字串長度都一樣的狀況
3. 當字串長度不同時可能要補空格

C.知其不可微而微之(Pseudo Differentiable)

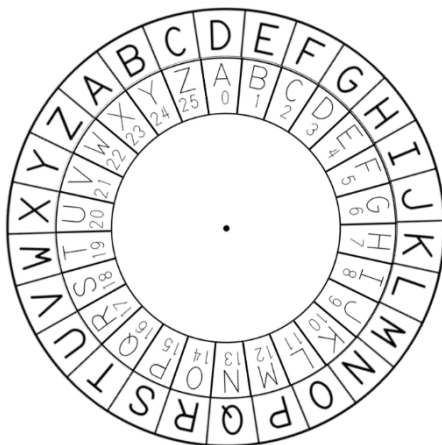
1. 這題就只是數學能力或者說閱讀能力而已
2. 常數函數微分必為 0
3. 記得開 long long

D.No 1 can solve this problem

1. 如果都沒有 1 的答案嫌而易見
2. 先想看如果只有一個 1 的時候答案是什麼(也可以試試看窮舉)
3. 數字很大 long long 可能讀不下(當然也有子題)

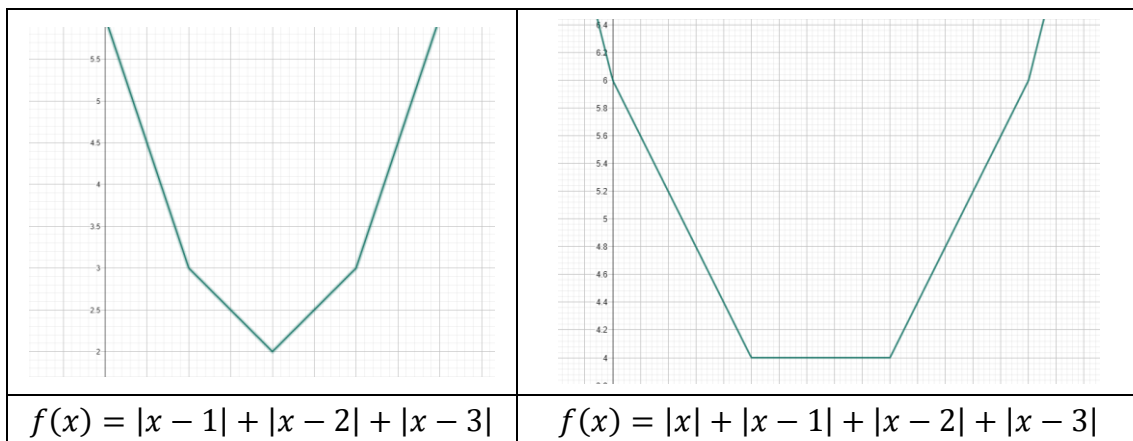
F.不時可憐地以遊戲為由被社員忽略的 CJ 題目(Ignorance)

1. $x = 0$ 根本沒變， $x = 2$ 是 CJ 上一模一樣的
2. 下面這張圖看你可以想到多少了



G. 驚奇的心動魔法(Magic)

1. 哇這紅頭髮怎麼這麼眼熟
2. 理論上這是高一數學的範圍，你應該要會所以我不想提示你
3. 參考一下這兩個圖



I. 巨震大師 (Matrix Master)

1. 看題目的英文名稱，顧名思義:矩陣大師，你可以用矩陣，前提是你要會
2. 不用矩陣也可以解，但程式比較難寫
3. 很多子題跟送分差不多，而且上次社課就已經洩題洩了 200 分

J. 計畫 C (Plan C)

1. 很明顯 Zhenzhe 並不會公式解，難道你不會嗎
2. 就算你不會公式解我相信你也有眼睛

至於 E 和 H 你問說為甚麼沒有提示

