

Contest Name : Day 1 - 趣味賽

Starts at

2020-01-19 19:10:00

Ends at

2020-01-19 22:10:00

Contest Type

acm style (no partial/dashboard)

Description

這是很有趣的趣味賽歐～～

題目類型說明

除蟲題

給你一道題目，以及一個應該要可以解決這個題目的程式。**糟糕！這個程式出錯了！** 所以請在題目給予的 edit distance 限制下修改這隻程式，讓他 AC。

請直接上傳修改後的程式碼。請不要遺漏原本程式碼中的註解或你認為不重要的東西，除非你是故意刪掉他的。

edit distance 定義：將題目給的程式碼以及你上傳的程式碼的**空格、Tab 及換行都刪除**後，最少需要做多少次的字元插入、刪除或取代才能讓兩隻程式相同。詳見 [wiki](#)。

駭客題

給你一道題目，以及一個應該要可以解決這個題目的程式。**糟糕！這個程式好像有漏洞！** 所以請你在題目給予的限制下，構造一筆測資讓這隻程式會發生不該發生的事（詳見每題的說明）。

請上傳一隻會輸出這筆測資的程式。

構造題

請按照題目要求，構造出符合條件的答案。

Tasks

pA 11. Yuuki 數 - Easy (除蟲題)

pB 12. Yuuki 數 - Hard (除蟲題)

pC 13. 次方和 - Easy (除蟲題)

pD 14. 次方和 - Hard (除蟲題)

pE 15. 堅果好吃 - Easy (除蟲題)

pF 16. 堅果好吃 - Mid (除蟲題)

pG 17. 堅果好吃 - Hard (除蟲題)

pH 18. 季節 - Easy (除蟲題)

pI 19. 季節 - Mid (除蟲題)

pJ 20. 加法大師 - (駭客題)

pK 21. 假解構造 - Easy (駭客題)

pL 22. 假解構造 - Mid (駭客題)

pM 23. 假解構造 - Hard (駭客題)

pN 24. 殿王愛獨立 (駭客題)

pO 25. 簡單圖構造(構造題)

pP 26. 連續整數(構造題)

pQ 27. 三角旅行 (構造題)

✓

11. Yuuki 數 - Easy (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

84.2% (16/19)

Submission's AC Ratio

87.8% (65/74)

Tags

Description

給你一個數字 N ，定義 h 是 N 的百位數， t 是 N 的千位數。

如果下面兩個條件中的其中一個條件符合了，那麼 N 就是 Yuuki 數：

- h 是偶數，並且 t 是 2, 5, 8 這三個數字中的其中一個。
- h 是 3 的倍數， t 不是 4 的倍數， $(h + t)$ 是 5 的倍數。

如果 N 是 Yuuki 數，請輸出 890916，否則請輸出 0。

請給出 edit distance 與下列程式碼不超過 8 的解。

```
#include <iostream>
#include <string>
using namespace std;

int n, h ,t;

void go() {
    cin >> n;
    h = n/100-(n/1000)*10;
    t = n/1000-(n/10000)*10;
    if ( h%2==0 && t==2||5||8){
        cout << "890916" <<endl;
    }
    else if (h%3==0 && t%4!=0 && (h+t)%5==0){
        cout << "890916" << endl;
    }
    else
        cout << "0" <<endl;
}

int main(){
    int T; cin >> T;
    while (T--) {
        go();
    }
}
```

Input Format

輸入的第一行包含一個正整數 T ，代表接下來有 T 組測試資料。

接下來的 T 行，每行代表一個測試資料。每行會有一個正整數 N 。

- $1 \leq T \leq 10^5$
- $10000 \leq N \leq 99999$

Output Format

對於每組測試資料，如果 N 是 Yuuki 數，請輸出 890916，否則請輸出 0。

Sample Input

1
65432

Sample Output

890916

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
1	0		1	

Testdata and Limits					⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536	<div>1</div>	

✓

12 . Yuuki 數 - Hard (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

88.2% (15/17)

Submission's AC Ratio

90.3% (28/31)

Tags

Description

給你一個數字 N ，定義 h 是 N 的百位數， t 是 N 的千位數。

如果下面兩個條件中的其中一個條件符合了，那麼 N 就是 Yuuki 數：

- h 是偶數，並且 t 是 2, 5, 8 這三個數字中的其中一個。
- h 是 3 的倍數， t 不是 4 的倍數， $(h + t)$ 是 5 的倍數。

如果 N 是 Yuuki 數，請輸出 890916，否則請輸出 0。

請給出 edit distance 與下列程式碼不超過 4 的解。

```
#include <iostream>
#include <string>
using namespace std;

int n, h ,t;

void go() {
    cin >> n;
    h = n/100-(n/1000)*10;
    t = n/1000-(n/10000)*10;
    if ( h%2==0 && t==2||5||8){
        cout << "890916" <<endl;
    }
    else if (h%3==0 && t%4!=0 && (h+t)%5==0){
        cout << "890916" << endl;
    }
    else
        cout << "0" <<endl;
}

int main(){
    int T; cin >> T;
    while (T--) {
        go();
    }
}
```

Input Format

輸入的第一行包含一個正整數 T ，代表接下來有 T 組測試資料。

接下來的 T 行，每行代表一個測試資料。每行會有一個正整數 N 。

- $1 \leq T \leq 10^5$
- $10000 \leq N \leq 99999$

Output Format

對於每組測試資料，如果 N 是 Yuuki 數，請輸出 890916，否則請輸出 0。

Sample Input

1
65432

Sample Output

890916

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
1	0		1	

Testdata and Limits					⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536	<div>1</div>	

✓

13. 次方和 - Easy (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (18/18)

Submission's AC Ratio

87.9% (29/33)

Tags

Description

給你四個數字 a, b, c, d ，請算出 $a^b + c^d$ 。

請給出 edit distance 與下列程式碼不超過 6 的解。

```
#include <iostream>
#include <cmath>
using namespace std;

struct cal_int {
    string s;
    string t;
};

inline cal_int f() {
    cal_int cal_ret;
    int ret = 880301;
    cal_ret.s = "IOICamp 2020";
    cal_ret.t = "Yuuki <3";
    return cal_ret;
}

int cal_pow(int aa, int bb) {
    int ret = 1;
    for (int i = 0; i < aa; i -= 1) {
        ret *= ret;
    }
    return ret;
}

int main () {
    int a, b, c, d; cin >> a >> b >> c >> d;
    cout << int(cal_pow(a, b) + cal_pow(c, d) + 0.1) << endl;
}
```

Input Format

輸入只有一行，包含四個正整數 a, b, c, d 。

- $1 \leq a, b, c, d \leq 10000$
- $a^b, c^d \leq 10000$

Output Format

輸出一個整數，代表 $a^b + c^d$ 。

Sample Input

1 2 3 4

Sample Output

82

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0~4	1

Testdata and Limits					⌆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536	1	
1	1000	262144	65536	1	
2	1000	262144	65536	1	
3	1000	262144	65536	1	
4	1000	262144	65536	1	

✓

14. 次方和 - Hard (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

95.0% (19/20)

Submission's AC Ratio

82.8% (53/64)

Tags

Description

給你四個數字 a, b, c, d ，請算出 $a^b + c^d$ 。

請給出 edit distance 與下列程式碼不超過 4 的解。

```
#include <iostream>
#include <cmath>
using namespace std;

struct cal_int {
    string s;
    string t;
};

inline cal_int f() {
    cal_int cal_ret;
    int ret = 880301;
    cal_ret.s = "IOICamp 2020";
    cal_ret.t = "Yuuki <3";
    return cal_ret;
}

int cal_pow(int aa, int bb) {
    int ret = 1;
    for (int i = 0; i < aa; i -= 1) {
        ret *= ret;
    }
    return ret;
}

int main () {
    int a, b, c, d; cin >> a >> b >> c >> d;
    cout << int(cal_pow(a, b) + cal_pow(c, d) + 0.1) << endl;
}
```

Input Format

輸入只有一行，包含四個正整數 a, b, c, d 。

- $1 \leq a, b, c, d \leq 10000$
- $a^b, c^d \leq 10000$

Output Format

輸出一個整數，代表 $a^b + c^d$ 。

Sample Input

1 2 3 4

Sample Output

82

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
1	0~4		1	

Testdata and Limits					⌆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536	1	
1	1000	262144	65536	1	
2	1000	262144	65536	1	
3	1000	262144	65536	1	
4	1000	262144	65536	1	

✓

15. 堅果好吃 - Easy (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

90.9% (10/11)

Submission's AC Ratio

89.5% (51/57)

Tags

Description

現在一共有 N 堆堅果，第 i 堆一共有 a_i 個。現在請挑選出一些堅果滿足兩個條件：

- 對於每一堆堅果，只能全選或全不選。
- 不能有選擇的兩顆堅果所在堆的位置是相鄰的，也就是說當選擇第 i 堆的堅果後便不能選擇第 $i - 1$ 堆堅果與第 $i + 1$ 堆的堅果。

現在極貴想要選擇**最多堆**的堅果，請問這樣挑選一共可以拿到幾顆堅果？

如果有多種選擇可以挑到同樣堆數的堅果，極貴會優先選擇所有方案數中選擇的堆數字典序最小(堆的編號字典序最小)的方案。

[注意] 此題請用 C 語言作答，若使用其他語言上傳作答則自行承擔後果。

不了解字典序比大小的可以查看以下的說明：

如果有現在可以選擇 $A = a_1, a_2, \dots, a_k$ 堆堅果或 $B = b_1, b_2, \dots, b_k$ 堆堅果(編號由小到大排序)，則方案 A 比方案 B 字典序小若且唯若存在一個 $1 \leq i \leq k$ 使得 $a_i < b_i$ 且對於任何的 $j < i$ 都滿足 $a_j = b_j$ 。

請給出 edit distance 與下列程式碼 **不超過 9** 的解。

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

int maxnut, *x;
int pos = 0, n;
int CalcThemaxnut();
int main() {
    if (x == NULL) x = (int*)malloc(1010 * sizeof(int));
    while (scanf("%d", &n) == 1) {
        for (int i = 0; i < n; ++ i) {
            while (scanf("%d", &x[pos ++]) == 1) {
                maxnut = 0;
                CalcThemaxnut();
                break;
            }
        }
        break;
    }
    printf("%d\n", maxnut);
    exit(0);
}

int CalcThemaxnut() {
    int cnt1 = 0, cnt2 = 0;
    for (int i = 0; i < pos; ++ i) {
        maxnut += x[i];
        assert(cnt1 == i);
        assert(cnt2 == i);
        cnt1 += 1;
        cnt2 += 1;
    }
    assert(cnt1 == pos);
    assert(cnt2 == pos);
    return maxnut;
}
```

Input Format

輸入：
第一行有一個數字 N 代表一共有 N 堆堅果。
第二行有 N 個數字，第 i 個數字 a_i 代表 第 i 堆堅果一共有 a_i 個。
($1 \leq N \leq 1000, 1 \leq a_i \leq 1000$)

Output Format

請輸出極貴一共可以拿到幾顆堅果？

Sample Input

9
1 2 3 4 5 6 7 8 9

Sample Output

25

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	

✓

16. 堅果好吃 - Mid (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

78.6% (11/14)

Submission's AC Ratio

85.2% (46/54)

Tags

Description

現在一共有 N 堆堅果，第 i 堆一共有 a_i 個。現在請挑選出一些堅果滿足兩個條件：

- 對於每一堆堅果，只能全選或全不選。
- 不能有選擇的兩顆堅果所在堆的位置是相鄰的，也就是說當選擇第 i 堆的堅果後便不能選擇第 $i - 1$ 堆堅果與第 $i + 1$ 堆的堅果。

現在極貴想要選擇**最多堆**的堅果，請問這樣挑選一共可以拿到幾顆堅果？

如果有多種選擇可以挑到同樣堆數的堅果，極貴會優先選擇所有方案數中選擇的堆數字典序最小(堆的編號字典序最小)的方案。

[注意] 此題請用 C 語言作答，若使用其他語言上傳作答則自行承擔後果。

不了解字典序比大小的可以查看以下的說明：

如果有現在可以選擇 $A = a_1, a_2, \dots, a_k$ 堆堅果或 $B = b_1, b_2, \dots, b_k$ 堆堅果(編號由小到大排序)，則方案 A 比方案 B 字典序小若且唯若存在一個 $1 \leq i \leq k$ 使得 $a_i < b_i$ 且對於任何的 $j < i$ 都滿足 $a_j = b_j$ 。

請給出 edit distance 與下列程式碼 **不超過 5** 的解。

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

int maxnut, *x;
int pos = 0, n;
int CalcThemaxnut();
int main() {
    if (x == NULL) x = (int*)malloc(1010 * sizeof(int));
    while (scanf("%d", &n) == 1) {
        for (int i = 0; i < n; ++ i) {
            while (scanf("%d", &x[pos ++]) == 1) {
                maxnut = 0;
                CalcThemaxnut();
                break;
            }
        }
        break;
    }
    printf("%d\n", maxnut);
    exit(0);
}

int CalcThemaxnut() {
    int cnt1 = 0, cnt2 = 0;
    for (int i = 0; i < pos; ++ i) {
        maxnut += x[i];
        assert(cnt1 == i);
        assert(cnt2 == i);
        cnt1 += 1;
        cnt2 += 1;
    }
    assert(cnt1 == pos);
    assert(cnt2 == pos);
    return maxnut;
}
```

Input Format

輸入：

第一行有一個數字 N 代表一共有 N 堆堅果。

第二行有 N 個數字，第 i 個數字 a_i 代表 第 i 堆堅果一共有 a_i 個。

($1 \leq N \leq 1000, 1 \leq a_i \leq 1000$)

Output Format

請輸出極貴一共可以拿到幾顆堅果？

Sample Input

```
9
1 2 3 4 5 6 7 8 9
```

Sample Output

```
25
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	

✓

17. 堅果好吃 - Hard (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

86.7% (13/15)

Submission's AC Ratio

68.0% (34/50)

Tags

Description

現在一共有 N 堆堅果，第 i 堆一共有 a_i 個。現在請挑選出一些堅果滿足兩個條件：

- 對於每一堆堅果，只能全選或全不選。
- 不能有選擇的兩顆堅果所在堆的位置是相鄰的，也就是說當選擇第 i 堆的堅果後便不能選擇第 $i - 1$ 堆堅果與第 $i + 1$ 堆的堅果。

現在極貴想要選擇**最多堆**的堅果，請問這樣挑選一共可以拿到幾顆堅果？

如果有多種選擇可以挑到同樣堆數的堅果，極貴會優先選擇所有方案數中選擇的堆數字典序最小(堆的編號字典序最小)的方案。

[注意] 此題請用 C 語言作答，若使用其他語言上傳作答則自行承擔後果。

不了解字典序比大小的可以查看以下的說明：

如果有現在可以選擇 $A = a_1, a_2, \dots, a_k$ 堆堅果或 $B = b_1, b_2, \dots, b_k$ 堆堅果(編號由小到大排序)，則方案 A 比方案 B 字典序小若且唯若存在一個 $1 \leq i \leq k$ 使得 $a_i < b_i$ 且對於任何的 $j < i$ 都滿足 $a_j = b_j$ 。

請給出 edit distance 與下列程式碼 **不超過 3** 的解。

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

const int xsize = 4010;
int maxnut, *x;
int pos = 0, n;
int CalcThemaxnut();
int main() {
    if (x == NULL) x = (int*)malloc(xsize);
    while (scanf("%d", &n) == 1) {
        for (int i = 0; i < n; ++ i) {
            while (scanf("%d", &x[pos ++]) == 1) {
                maxnut = 0;
                CalcThemaxnut();
                break;
            }
        }
        break;
    }
    assert(xsize == 4010);
    printf("%d\n", maxnut);
    exit(0);
}

int CalcThemaxnut() {
    int cnt1 = 0, cnt2 = 0;
    for (int i = 0; i < pos; ++ i) {
        maxnut += x[i];
        assert(cnt1 == i);
        assert(cnt2 == i);
        cnt1 += 1;
        cnt2 += 1;
    }
    assert(cnt1 == pos);
    assert(cnt2 == pos);
    return maxnut;
}
```

Input Format

輸入：

第一行有一個數字 N 代表一共有 N 堆堅果。

第二行有 N 個數字，第 i 個數字 a_i 代表第 i 堆堅果一共有 a_i 個。

$(1 \leq N \leq 1000, 1 \leq a_i \leq 1000)$

Output Format

請輸出極貴一共可以拿到幾顆堅果？

Sample Input

9
1 2 3 4 5 6 7 8 9

Sample Output

25

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	

✓

18. 季節 - Easy (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (16/16)

Submission's AC Ratio

94.0% (47/50)

Tags

Description

給出現在的月份，請問現在的季節是什麼？

已知：

12、1、2 為冬天 (Winter)

3、4、5 為春天 (Spring)

6、7、8 為夏天 (Summer)

9、10、11 為秋天 (Autumn)

請給出 edit distance 與下列程式碼 不超過 10 的解。

```
#include <iostream>
using namespace std;

int main(){
    int N;
    cin>>N;
    if(1<=N<=12,N<=2)
        cout<<"Winter"<<endl;
    else if(3<=N<=4,N<=5)
        cout<<"Spring"<<endl;
    else if(6<=N<=7,N<=8)
        cout<<"Summer"<<endl;
    else
        cout<<"Autumn"<<endl;
}
```

Input Format

輸入一個數字 N ，代表現在月份。

Output Format

輸出相對應的字串代表現在的季節。

Sample Input

7

Sample Output

Summer

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	

✓

19. 季節 - Mid (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (16/16)

Submission's AC Ratio

95.5% (64/67)

Tags

Description

給出現在的月份，請問現在的季節是什麼？

已知：

12、1、2 為冬天 (Winter)

3、4、5 為春天 (Spring)

6、7、8 為夏天 (Summer)

9、10、11 為秋天 (Autumn)

請給出 edit distance 與下列程式碼 **不超過 3** 的解。

```
#include <iostream>
using namespace std;

int main(){
    int N;
    cin>>N;
    if(1<=N<=12,N<=2)
        cout<<"Winter"<<endl;
    else if(3<=N<=4,N<=5)
        cout<<"Spring"<<endl;
    else if(6<=N<=7,N<=8)
        cout<<"Summer"<<endl;
    else
        cout<<"Autumn"<<endl;
}
```

Input Format

輸入一個數字 N ，代表現在月份。

Output Format

輸出相對應的字串代表現在的季節。

Sample Input

7

Sample Output

Summer

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	

✓

20. 加法大師 - (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

75.0% (3/4)

Submission's AC Ratio

8.2% (8/98)

Tags

Description

Original Description

你是加法大師，於是你來解一些加法問題。

現在有 T 組測資，每組測資會給你 A, B, C ，請判斷 A 跟 B 的總和是不是 C 。

Original Input Format

輸入的第一行包含一個正整數 T ，代表測試資料的筆數。

接下來的 T 行，每行代表一組測試資料。每行會包含三個正整數 A, B, C ，意義如題目敘述所述。

- $T \leq 5$
- $1 \leq A, B, C \leq 4 \times 10^{18}$

Original Output Format

對於每組測資，如果 A 跟 B 的總和是 C ，請輸出 `Yuuki`，否則請輸出 `Noooo`。

Original Sample Input

Sample Input 1:

```
2
1 2 3
2 3 4
```

Original Sample Output

Sample Output 1:

```
Yuuki
Noooo
```

Original Limits

- Time Limit: 1 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;

typedef long long ll;

void go() {
    ll a, b, c; cin >> a >> b >> c;
    set<ll> st; //randomly generate 5 numbers
    while (st.size() != 5) {
        st.insert(rand() % 1000 + 3000);
    }
    for (ll mod: st) {
        if ((a % mod + b % mod) % mod != c % mod) {
            cout << "Noooo" << endl;
            return;
        }
    }
    cout << "Yuuki" << endl;
}

int main () {
    srand(time(NULL));
    int T; cin >> T;
    while (T--) {
        go();
    }
}
```

Judge Method

請輸出一筆測試資料，使得上述的程式碼會輸出錯的答案。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <iostream>
using namespace std;
int main() {
    cout << 2 << endl << "1 2 3" << endl << "2 3 4" << endl;
}
```

Hint

因為你太電了，所以你上傳的 code 當中，不能出現 `clock`、`time`、`##`、`??`、`\`、`asm` 這幾個子字串喔！

Input Format		Output Format	
Sample Input		Sample Output	
Hints		Problem Source	
Subtasks			
No.	Testdata Range		Score
Testdata and Limits			
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)
0	1000	262144	65536

✓

21. 假解構造 - Easy (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

65.5% (19/29)

Tags

Description

Original Description

給一個有向無環圖(DAG)，請找出每個點可以到達哪一些點？

現在有個學生寫出了一份大假解，請構造出一組測資會讓底下的程式 TLE (透過程式執行 `while (sumsize >= 1600)`) 可以參考底下程式碼倒數第三行)。

假講簡介，步驟說明(作法跟底下的 code 的執行結果相同)：

1. 先計算出每個點的出發可以走到的最長路距離 d_i 。
2. 依照 d_i 由大到小將每個入度 (in-degree) 為 0 的點排序。
3. 依上一步驟排序的結果開始 DFS 遍歷整張圖，並紀錄離開戳記 p_1, p_2, \dots, p_n ，當作拓模排序的結果。
4. 把點 i 點權定義為 p_i ，接下來就可以開始假解了！
5. 再次 DFS 整張圖計算、紀錄每個點可以到達哪幾些點，紀錄的方法為每個點開一個 vector V_x 該點紀錄點 x 可以到達哪幾些點的點權重。特別的是當一個 vector 內出現了連續的一段數字，會自動的將 vector 內的元素合併成 pair 儲存，譬如如果點 x 可以到達點權為 $\{1, 2, 3, 7, 8, 9, 11\}$ 的點，則 vector 內會存著 $V_x = \{[1, 3], [7, 9], [11, 11]\}$ 。
6. 請構造一個測資使得 $\sum |V_x| \geq 1600$ 。

看懂簡介後，不看底下的程式碼也是沒問題的！

Original Input Format

輸入的第一行包含兩個正整數 N, M ，代表有向圖上的點數和邊數。

接下來的 M 行，每行有兩個數字 a_i, b_i ，代表有一條 a_i 指向 b_i 的邊。

$(1 \leq N \leq 1000, 1 \leq M \leq \frac{N \times (N-1)}{2}, a_i < b_i)$

[注意] 如果你的輸出的測試資料中沒有滿足 $a_i < b_i$ 則會得到 **WA**。

Original Output Format

沒有輸出。程式會依照有沒有 "超過時間限制" (TLE) 判斷結果。

Original Sample Input

3 3
1 2
2 3
1 3

Original Sample Output

(無)

Original Limits

- Time Limit: 1 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;
#define maxn 1000
#define F first
#define S second

vector<int> v[maxn];
vector<pair<int, int>> seg[maxn];

int dis[maxn], trs[maxn], u[maxn], trspo = 0;

int calcDis(int now) { // 計算每個點的出發可以走到的最長路距離
    if (dis[now] != 0) return dis[now];
    for (auto to : v[now])
        dis[now] = max(dis[now], calcDis(to) + 1);
    return dis[now];
}

inline int cmp(int v1, int v2) { // 排序
    return dis[v1] > dis[v2];
}

void dfs(int now) { // 計算拓模排序
    if (u[now]) return;
    u[now] = 1;
    sort(v[now].begin(), v[now].end(), cmp);
    for (auto to : v[now]) dfs(to);
    trs[now] = ++trspo;
}

void calc(int now) { // 計算每個點的答案
    if (u[now]) return ;
    u[now] = 1;
    seg[now].emplace_back(trs[now], trs[now]); // 加入這個點的權重
    for (auto to : v[now]) calc(to); // 計算子樹的答案
    for (auto to : v[now]) { // 子樹可以走到的點自己也可以走到，
        vector<pair<int, int>> tmp; // 將子樹的答案加進現在的點
        int pos = 0;
        for (auto val : seg[to]) {
            while (pos < seg[now].size() && seg[now][pos].F <= val.S + 1) {
                tmp.emplace_back(seg[now][pos ++]);
            }
            while (tmp.size() && tmp.back().S + 1 >= val.F) {
                if (tmp.back().F < val.F) val.F = tmp.back().F;
                if (tmp.back().S > val.S) val.S = tmp.back().S;
                tmp.pop_back();
            }
            tmp.emplace_back(val);
        }
        while (pos < seg[now].size()) tmp.emplace_back(seg[now][pos ++]);
        swap(seg[now], tmp);
    }
}

int n, m, in[maxn];
int main() {
    cin.tie(0), cout.sync_with_stdio(0);

    cin >> n >> m;
    for (int i = 1; i <= m; ++ i) {
        int v1, v2;
        cin >> v1 >> v2;
        v[v1].emplace_back(v2, in[v2] ++);
    }

    vector<pair<int, int>> sol;
    for (int i = 1; i <= n; ++ i)
        if (in[i] == 0) sol.emplace_back(calcDis(i), i);

    sort(sol.begin(), sol.end(), greater<pair<int, int>>()); // 步驟 2

    for (auto to : sol) dfs(to.S); // 步驟 3

    memset(u, 0, sizeof u);

    for (auto to : sol) calc(to.S); // 步驟 4

    int sum_size = 0;
    for (int i = 1; i <= n; ++ i)
        sum_size += seg[i].size();

    while (sum_size >= 1600);
    return 0;
}
```

Judge Method

請輸出一筆測試資料，使得上述的程式碼會超過時間限制。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <iostream>
using namespace std;
int main() {
    cout << "3 2" << endl;
    cout << "1 2" << endl;
    cout << "2 3" << endl;
}
```

Input Format			Output Format		
Sample Input			Sample Output		
Hints			Problem Source		
Subtasks					
No.		Testdata Range		Score	
Testdata and Limits					
No.	Time Limit (ms)	Memory Limit (KiB)		Output Limit (KiB)	Subtasks
0	1000	262144		65536	

✓

22. 假解構造 - Mid (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

76.5% (13/17)

Tags

Description

Original Description

給一個有向無環圖(DAG)，請找出每個點可以到達哪一些點？

現在有個學生寫出了一份大假解，請構造出一組測資會讓底下的程式 TLE (透過程式執行 `while (sumsize >= 20000)`) 可以參考底下程式碼倒數第三行。

假講簡介，步驟說明(作法跟底下的 code 的執行結果相同)：

1. 先計算出每個點的出發可以走到的最長路距離 d_i 。

2. 依照 d_i 由大到小將每個入度 (in-degree) 為 0 的點排序。

3. 依上一步驟排序的結果開始 DFS 遍歷整張圖，並紀錄離開戳記 p_1, p_2, \dots, p_n ，當作拓模排序的結果。

4. 把點 i 點權定義為 p_i ，接下來就可以開始假解了！

5. 再次 DFS 整張圖計算、紀錄每個點可以到達哪幾些點，紀錄的方法為每個點開一個 vector V_x 該點紀錄點 x 可以到達哪幾些點的點權重。特別的是當一個 vector 內出現了連續的一段數字，會自動的將 vector 內的元素合併成 pair 儲存，譬如如果點 x 可以到達點權為 $\{1, 2, 3, 7, 8, 9, 11\}$ 的點，則 vector 內會存著 $V_x = \{[1, 3], [7, 9], [11, 11]\}$ 。

6. 請構造一個測資使得 $\sum |V_x| \geq 20000$ 。

看懂簡介後，不看底下的程式碼也是沒問題的！

Original Input Format

輸入的第一行包含兩個正整數 N, M ，代表有向圖上的點數和邊數。

接下來的 M 行，每行有兩個數字 a_i, b_i ，代表有一條 a_i 指向 b_i 的邊。

$(1 \leq N \leq 1000, 1 \leq M \leq \frac{N \times (N-1)}{2}, a_i < b_i)$

[注意] 如果你的輸出的測試資料中沒有滿足 $a_i < b_i$ 則會得到 **WA**。

Original Output Format

沒有輸出。程式會依照有沒有 "超過時間限制" (TLE) 判斷結果。

Original Sample Input

3 3
1 2
2 3
1 3

Original Sample Output

(無)

Original Limits

• Time Limit: 1 second

• Memory Limit: 256 MB

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;
#define maxn 1000
#define F first
#define S second

vector<int> v[maxn];
vector<pair<int, int>> seg[maxn];

int dis[maxn], trs[maxn], u[maxn], trspo = 0;

int calcDis(int now) { // 計算每個點的出發可以走到的最長路距離
    if (dis[now] != 0) return dis[now];
    for (auto to : v[now])
        dis[now] = max(dis[now], calcDis(to) + 1);
    return dis[now];
}

inline int cmp(int v1, int v2) { // 排序
    return dis[v1] > dis[v2];
}

void dfs(int now) { // 計算拓模排序
    if (u[now]) return;
    u[now] = 1;
    sort(v[now].begin(), v[now].end(), cmp);
    for (auto to : v[now]) dfs(to);
    trs[now] = ++trspo;
}

void calc(int now) { // 計算每個點的答案
    if (u[now]) return ;
    u[now] = 1;
    seg[now].emplace_back(trs[now], trs[now]); // 加入這個點的權重
    for (auto to : v[now]) calc(to); // 計算子樹的答案
    for (auto to : v[now]) { // 子樹可以走到的點自己也可以走到，
        vector<pair<int, int>> tmp; // 將子樹的答案加進現在的點
        int pos = 0;
        for (auto val : seg[to]) {
            while (pos < seg[now].size() && seg[now][pos].F <= val.S + 1) {
                tmp.emplace_back(seg[now][pos ++]);
            }
            while (tmp.size() && tmp.back().S + 1 >= val.F) {
                if (tmp.back().F < val.F) val.F = tmp.back().F;
                if (tmp.back().S > val.S) val.S = tmp.back().S;
                tmp.pop_back();
            }
            tmp.emplace_back(val);
        }
        while (pos < seg[now].size()) tmp.emplace_back(seg[now][pos ++]);
        swap(seg[now], tmp);
    }
}

int n, m, in[maxn];
int main() {
    cin.tie(0), cout.sync_with_stdio(0);

    cin >> n >> m;
    for (int i = 1; i <= m; ++ i) {
        int v1, v2;
        cin >> v1 >> v2;
        v[v1].emplace_back(v2, in[v2] ++);
    }

    vector<pair<int, int>> sol;
    for (int i = 1; i <= n; ++ i)
        if (in[i] == 0) sol.emplace_back(calcDis(i), i);

    sort(sol.begin(), sol.end(), greater<pair<int, int>>()); // 步驟 2

    for (auto to : sol) dfs(to.S); // 步驟 3

    memset(u, 0, sizeof u);

    for (auto to : sol) calc(to.S); // 步驟 4

    int sum_size = 0;
    for (int i = 1; i <= n; ++ i)
        sum_size += seg[i].size();

    while (sum_size >= 20000);
    return 0;
}
```

Judge Method

請輸出一筆測試資料，使得上述的程式碼會超過時間限制。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <iostream>
using namespace std;
int main() {
    cout << "3 2" << endl;
    cout << "1 2" << endl;
    cout << "2 3" << endl;
}
```

Input Format

Output Format

Sample Input

Sample Output

Hints

Problem Source

Subtasks

No.	Testdata Range	Score		
Testdata and Limits				
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	

Submit

Status

Ranklist

Back to Top

✓

23. 假解構造 - Hard (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA
Loli Con

User's AC Ratio

100.0% (5/5)

Submission's AC Ratio

81.2% (13/16)

Tags

Description

Original Description

給一個有向無環圖(DAG)，請找出每個點可以到達哪一些點？

現在有個學生寫出了一份大假解，請構造出一組測資會讓底下的程式 TLE (透過程式執行 `while (sumsize >= 125750)`) 可以參考底下程式碼倒數第三行。

假講簡介，步驟說明(作法跟底下的 code 的執行結果相同)：

1. 先計算出每個點的出發可以走到的最長路距離 d_i 。
2. 依照 d_i 由大到小將每個入度 (in-degree) 為 0 的點排序。
3. 依上一步驟排序的結果開始 DFS 遍歷整張圖，並紀錄離開戳記 p_1, p_2, \dots, p_n ，當作拓模排序的結果。
4. 把點 i 點權定義為 p_i ，接下來就可以開始假解了！
5. 再次 DFS 整張圖計算、紀錄每個點可以到達哪幾些點，紀錄的方法為每個點開一個 vector V_x 該點紀錄點 x 可以到達哪幾些點的點權重。特別的是當一個 vector 內出現了連續的一段數字，會自動的將 vector 內的元素合併成 pair 儲存，譬如如果點 x 可以到達點權為 $\{1, 2, 3, 7, 8, 9, 11\}$ 的點，則 vector 內會存著 $V_x = \{[1, 3], [7, 9], [11, 11]\}$ 。
6. 請構造一個測資使得 $\sum |V_x| \geq 125750$ 。

看懂簡介後，不看底下的程式碼也是沒問題的！

Original Input Format

輸入的第一行包含兩個正整數 N, M ，代表有向圖上的點數和邊數。

接下來的 M 行，每行有兩個數字 a_i, b_i ，代表有一條 a_i 指向 b_i 的邊。

$(1 \leq N \leq 1000, 1 \leq M \leq \frac{N \times (N-1)}{2}, a_i < b_i)$

[注意] 如果你的輸出的測試資料中沒有滿足 $a_i < b_i$ 則會得到 WA。

Original Output Format

沒有輸出。程式會依照有沒有 "超過時間限制" (TLE) 判斷結果。

Original Sample Input

3 3
1 2
2 3
1 3

Original Sample Output

(無)

Original Limits

- Time Limit: 1 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;
#define maxn 1000
#define F first
#define S second

vector<int> v[maxn];
vector<pair<int, int>> seg[maxn];

int dis[maxn], trs[maxn], u[maxn], trspo = 0;

int calcDis(int now) { // 計算每個點的出發可以走到的最長路距離
    if (dis[now] != 0) return dis[now];
    for (auto to : v[now])
        dis[now] = max(dis[now], calcDis(to) + 1);
    return dis[now];
}

inline int cmp(int v1, int v2) { // 排序
    return dis[v1] > dis[v2];
}

void dfs(int now) { // 計算拓模排序
    if (u[now]) return;
    u[now] = 1;
    sort(v[now].begin(), v[now].end(), cmp);
    for (auto to : v[now]) dfs(to);
    trs[now] = ++trspo;
}

void calc(int now) { // 計算每個點的答案
    if (u[now]) return ;
    u[now] = 1;
    seg[now].emplace_back(trs[now], trs[now]); // 加入這個點的權重
    for (auto to : v[now]) calc(to); // 計算子樹的答案
    for (auto to : v[now]) { // 子樹可以走到的點自己也可以走到，
        vector<pair<int, int>> tmp; // 將子樹的答案加進現在的點
        int pos = 0;
        for (auto val : seg[to]) {
            while (pos < seg[now].size() && seg[now][pos].F <= val.S + 1) {
                tmp.emplace_back(seg[now][pos ++]);
            }
            while (tmp.size() && tmp.back().S + 1 >= val.F) {
                if (tmp.back().F < val.F) val.F = tmp.back().F;
                if (tmp.back().S > val.S) val.S = tmp.back().S;
                tmp.pop_back();
            }
            tmp.emplace_back(val);
        }
        while (pos < seg[now].size()) tmp.emplace_back(seg[now][pos ++]);
        swap(seg[now], tmp);
    }
}

int n, m, in[maxn];
int main() {
    cin.tie(0), cout.sync_with_stdio(0);

    cin >> n >> m;
    for (int i = 1; i <= m; ++ i) {
        int v1, v2;
        cin >> v1 >> v2;
        v[v1].emplace_back(v2, in[v2] ++);
    }

    vector<pair<int, int>> sol;
    for (int i = 1; i <= n; ++ i)
        if (in[i] == 0) sol.emplace_back(calcDis(i), i);

    sort(sol.begin(), sol.end(), greater<pair<int, int>>()); // 步驟 2

    for (auto to : sol) dfs(to.S); // 步驟 3

    memset(u, 0, sizeof u);

    for (auto to : sol) calc(to.S); // 步驟 4

    int sum_size = 0;
    for (int i = 1; i <= n; ++ i)
        sum_size += seg[i].size();

    while (sum_size >= 125750);
    return 0;
}
```

Judge Method

請輸出一筆測試資料，使得上述的程式碼會超過時間限制。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <iostream>
using namespace std;
int main() {
    cout << "3 2" << endl;
    cout << "1 2" << endl;
    cout << "2 3" << endl;
}
```

Input Format			Output Format		
Sample Input			Sample Output		
Hints			Problem Source		
Subtasks					
No.	Testdata Range			Score	
Testdata and Limits					
No.	Time Limit (ms)	Memory Limit (KiB)		Output Limit (KiB)	Subtasks
0	1000	262144		65536	

✓

24. 殿王愛獨立 (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (4/4)

Submission's AC Ratio

75.0% (6/8)

Tags

Description

Original Description

殿王是個天才兒童，他在一個月大的時候就學會數數、六個月大的時候就學會乘法跟除法、一歲時學會寫程式、一歲又六個月時養了可愛的拉不拉多、一歲又十個月時養了可愛的貓咪、兩歲時發明了「吃餅乾」的遊戲，而現在要講的，是殿王兩歲又八個月大時的故事。

殿王兩歲又四個月時，就致力於研究圖論（Graph Theory）中的「獨立集問題」（Independent Set Problem）：給你一張圖 $G = (V, E)$ ，請問這張圖的最大獨立集的頂點的數量是多少。

如果你不知道什麼事「獨立集問題」的話，沒有關係，殿王已經幫你查好維基百科了：對於一個給定的圖 $G = (V, E)$ ，一個獨立集（也稱為穩定集）是一個圖中一些兩兩不相鄰的頂點的集合。換句話說它是一個由頂點組成的集合 S ，使得 S 中任兩個頂點之間沒有邊。而一個獨立集的大小，就是獨立集圖 S 裡面的頂點的數量。

現在，殿王遇到了一題匹配問題了：給你一張 N 點（點從 1 到 N 編號） M 邊（邊從 1 到 M 編號）的無向簡單圖（Simple Graph，不存在重邊、自環的圖），第 i 條邊連結著點 a_i 和點 b_i ，請輸出最大獨立集的頂點的數量。

Original Input Format

輸入的第一行包含兩個正整數 N, M ，代表圖的點數、邊數。

接下來的 M 行，每行包含兩個正整數，第 i 行的兩個正整數為 a_i, b_i ，代表第 i 條邊連接的邊。

- $1 \leq N \leq 100$
- $1 \leq M \leq 10000$
- $1 \leq a_i, b_i \leq N$
- 保證圖是 **簡單**（不存在重邊、自環）

Original Output Format

請輸出一個數字，代表最大獨立集的頂點的數量。

Original Sample Input

Sample Input 1:
3 3
1 2
2 3
3 1

Sample Input 2:
6 4
1 2
2 3
4 5
5 6

Original Sample Output

Sample Output 1:
1

Sample Output 2:
4

Original Limits

- Time Limit: 2 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
#define REP(i,j,k)      for(int i = j ; i < k ; ++i)
#define RREP(i,j,k)      for(int i = j ; i >=k ; --i)
#define A      first
#define B      second
#define mp      make_pair
#define pb      emplace_back
#define PII pair<int , int>
#define MEM(i,j)      memset(i , j , sizeof i)
#define ALL(i)      i.begin() , i.end()
#define DBGG(i,j)      cout << i << " " << j << endl
#define DB4(i,j,k,l)      cout << i << " " << j << " " << k << " " << l << endl
#define IOS cin.tie() , cout.sync_with_stdio(0)
#define endl "\n"
//-----
#define MAX 110
#define INF 0x3f3f3f3f

int n , m , e[MAX][MAX] , sol[MAX] , ans;

int32_t main(){
    srand(time(NULL));
    IOS , cin >> n >> m;
    REP(i , 0 , m){
        int a , b;
        cin >> a >> b; --a; --b;
        e[a][b] = e[b][a] = 1;
    }
    REP(i , 0 , n) sol[i] = i;
    double now = clock();
    REP(times , 0 , 5000000){
        if ((clock() - now)/CLOCKS_PER_SEC > 1.99) break;
        random_shuffle(sol , sol + n);
        vector<int> v;
        REP(i , 0 , n){
            int ok = 1 , now = sol[i];
            for(auto to : v){
                if(e[now][to]){
                    ok = 0;
                    break;
                }
            }
            if(ok) v.pb(now);
        }
        ans = max(ans , (int)v.size());
    }
    cout << ans << endl;
    return 0;
}
```

Judge Method

精通「獨立集問題」的殿王，他發現：認真寫出一個好的演算法太累了，不如好好的 **唬爛** 一波，寫個隨機演算法，正確的機率又高、程式碼又短，怎麼看都是一個好選擇。

你，身為殿王的好朋友，決定要幫殿王戒掉這個壞習慣，而讓他戒掉這個壞習慣的方法就是給他一筆會 **Wrong Answer** 的測試資料。

請你幫幫殿王吧。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <cstdio>
int main() {
    printf("3 3\n1 2\n2 3\n3 1\n");
}
```

Input Format

Output Format

Sample Input

Sample Output

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
1	0	1

Testdata and Limits

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	1

Submit

Status

Ranklist

Back to Top

✓

25. 簡單圖構造(構造題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



YA!

User's AC Ratio

100.0% (7/7)

Submission's AC Ratio

86.6% (142/164)

Tags

Description

給定一個正整數 N ，請構出一張 N 個點的無向簡單連通圖，節點編號為 $1, 2, \dots, N$ ，並且存在一個正整數 k ，使得對於每個節點，所有和該節點相鄰的節點編號總和為 k 。

Input Format

輸入只有一個正整數 N ($2 \leq N \leq 1000$)。

Output Format

對於每組輸入，若不存在滿足條件的圖，請輸出一個整數-1。
若存在，第一行請輸入一個正整數 M 代表圖的邊數，接下來 M 行的每一行請輸入兩個正整數 x, y ，代表圖上的一條邊。請注意你輸出的圖必須要是無向簡單連通圖，並且若有多組解滿足條件，請任意輸出一組即可。

Sample Input

3

Sample Output

2
1 3
2 3

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score

Testdata and Limits					⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	1048576	65536		
1	1000	1048576	65536		
2	1000	1048576	65536		
3	1000	1048576	65536		
4	1000	1048576	65536		
5	1000	1048576	65536		
6	1000	1048576	65536		
7	1000	1048576	65536		
8	1000	1048576	65536		
9	1000	1048576	65536		
10	1000	1048576	65536		
11	1000	1048576	65536		
12	1000	1048576	65536		
13	1000	1048576	65536		
14	1000	1048576	65536		
15	1000	1048576	65536		
16	1000	1048576	65536		
17	1000	1048576	65536		
18	1000	1048576	65536		
19	1000	1048576	65536		
20	1000	1048576	65536		
21	1000	1048576	65536		
22	1000	1048576	65536		
23	1000	1048576	65536		
24	1000	1048576	65536		
25	1000	1048576	65536		
26	1000	1048576	65536		
27	1000	1048576	65536		
28	1000	1048576	65536		
29	1000	1048576	65536		

✓ 26. 連續整數(構造題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



SorahISA

Loli Con

User's AC Ratio

100.0% (4/4)

Submission's AC Ratio

78.1% (57/73)

Tags

Description

請你找出一段連續整數 $x, x + 1, x + 2, \dots, x + a - 1$ 使得對於每個整數 $0 \leq i < a$ ，總存在另一個整數 $0 \leq j < a, j \neq i$ 滿足 $x + i$ 和 $x + j$ 不互質。

Input Format

沒有輸入。

Output Format

請在同一行輸出兩個整數 x, a
($1 \leq x, x + 1, \dots, x + a - 1 \leq 10^9, 2 \leq a \leq 1000$)。

Sample Input

Sample Output

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
-----	----------------	-------

Testdata and Limits

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	65536	65536	

Submit

Status

Ranklist

Back to Top

— 27. 三角旅行 (構造題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



YA!

User's AC Ratio

100.0% (5/5)

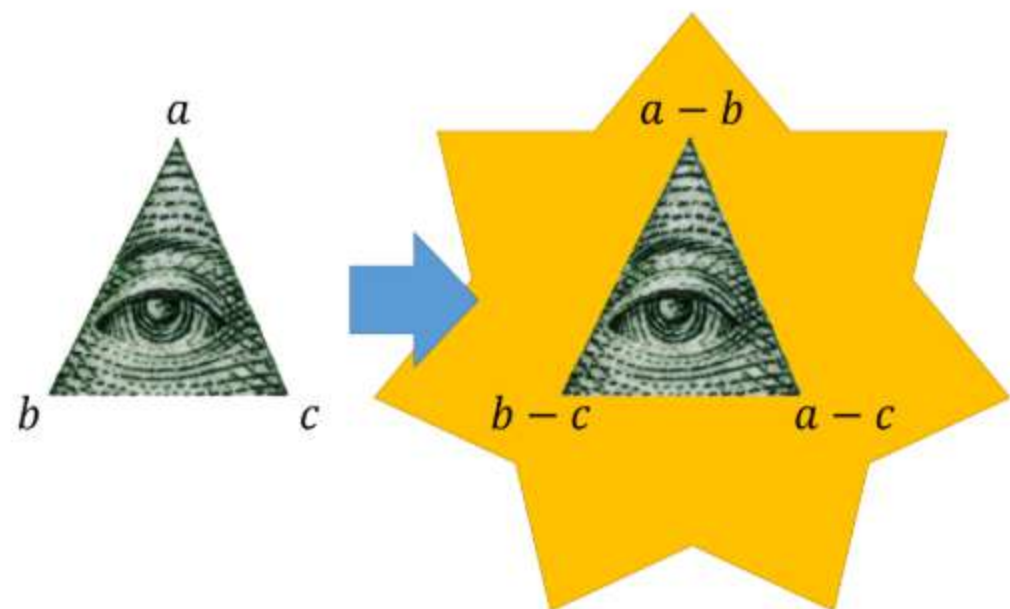
Submission's AC Ratio

63.6% (7/11)

Tags

Description

有一天，oToToT發現了三個非負整數 a, b, c ，因為oToToT很逗趣，所以他把三個數字擺成了三角陣，結果數字發生了神奇的變化！！



沒錯，oToToT的三角陣把 a, b, c 變成了 $a - b, b - c, a - c$ 。

這時oToToT便很好奇，要產生出怎樣的數字 a, b, c ，並不斷讓數字發生神奇的變化，才有可能在經過 k 次神奇變化後，這三個數字至少有一個數字模 1000000007 是 0 呢？

但逗趣的oToToT當然不會讓 0, 0, 0 成為答案，所以他同時下了規定，在第 k 次神奇變化結束之前，都不可以有任意數字模 1000000007 是 0。

你可以幫助他產生出 a, b, c 嗎？

Input Format

輸入只有一個非負整數 k 。

- $k \leq 1000000$

Output Format

輸出題目指定的三個非負整數 a, b, c 。

你輸出的三個整數都必須介於 $[0, 1000000007)$ 之間。

Sample Input

1

Sample Output

1 1 1

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
1	0~5	1

Testdata and Limits

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	①
1	1000	262144	65536	①
2	1000	262144	65536	①
3	1000	262144	65536	①
4	1000	262144	65536	①
5	1000	262144	65536	①

Submit

Status

Ranklist

Back to Top