

Tổng quan bài thi

STT	Tên bài	Dữ liệu vào	Bài làm	Giới hạn thời gian	Số điểm
1	Bình phương	Trong đề thi	binhphuong.txt	Không có	30
2	Hình chữ nhật	Trong đề thi	hcn.txt	Không có	30
3	Đoán số	ucnlib.h/ucnlib.pp	doanso.*	1 giây	40

Thí sinh sẽ thay dấu .* bằng .pas hoặc .cpp tùy theo bài làm bằng Pascal hoặc C++.

Bài 1: BÌNH PHƯƠNG

Bé Cam hôm nay đến lớp học Toán và được cô dạy về bình phương của một số. Cô cho Cam một bảng A kích thước $10^6 \times 10^6$ phần tử. Các dòng được đánh số từ 1 đến 10^6 theo chiều từ dưới lên trên, các cột được đánh số từ 1 đến 10^6 theo chiều từ trái sang phải. Phần tử nằm ở hàng i, cột j được kí hiệu là A_{ij} được tính bằng công thức $A_{ij} = i^2 + j^2$. Cô giáo tạo ra dãy B đánh số từ 1 đến 10^{12} gồm toàn bộ các phần tử của bảng A rồi sắp xếp theo thứ tự không giảm. Sau đó cô hỏi Cam xem số thứ K trong dãy B có giá trị là bao nhiêu? Lúc đầu cô đưa số K nhỏ nên Cam còn đếm được. Khi số K lớn lên, Cam không còn tính nhẩm trong đầu được nữa, Cam muốn các bạn thi Tin học trẻ hôm nay giúp bé tìm đáp án cho câu hỏi của cô giáo nhé.

	1	2	3	4	5	...
5	26	29	34	41	50	...
4	17	20	25	32	41	...
3	10	13	18	25	34	...
2	5	8	13	20	29	...
1	2	5	10	17	26	...
	1	2	3	4	5	...

Hình bên là ảnh Cam chụp góc trái dưới cùng bảng A.

Sau khi chuyển các phần tử của bảng A vào dãy B và sắp xếp theo thứ tự không giảm thì Cam có những phần tử đầu tiên của dãy B là {2, 5, 5, 8, 10, 10, 13, 13, 17, 17, 18...}

Test	K
1	9
2	17
3	26
4	100
5	12345
6	54321
7	20172017
8	123456789
9	9999999999
10	201720172017

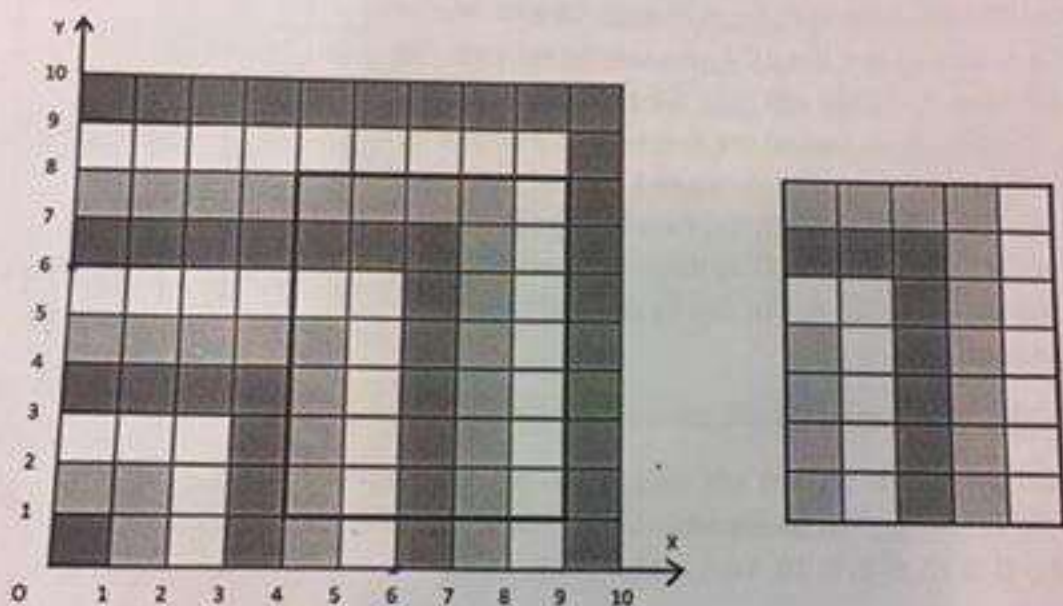
Khi cô hỏi K=5 thì Cam đã ngay lập tức đưa ra được đáp án là 10. Các bạn hãy giúp bé với những số K lớn hơn nhé.

Bạn sẽ nhận được một bảng dữ liệu đầu vào bên dưới gồm 10 dòng, tương ứng với 10 câu hỏi, mỗi câu gồm một số K. Bạn cần tạo file BINHPHUONG.TXT gồm 10 dòng tương ứng với 10 câu hỏi, mỗi dòng ghi 1 số nguyên dương duy nhất là số thứ K trong dãy B.

Bài 2: HÌNH CHỮ NHẬT

Giờ học toán tiếp theo, Cam vẽ một trục tọa độ Oxy lên giấy và dùng 3 màu đen, xám, trắng để tô màu góc phần tư thứ nhất. Đầu tiên Cam tô hình vuông có tọa độ góc trái trên $(0,1)$ và tọa độ góc phải dưới $(1,0)$ là màu đen. Tiếp theo bé tô những ô kề đỉnh với ô đen bằng màu xám, sau đó lại tô những ô kề đỉnh với ô xám mà chưa được tô bằng màu trắng, sau đó lại tô những ô kề đỉnh với ô màu trắng mà chưa được tô bằng màu đen và cứ tiếp tục như vậy... Sau đó Cam sẽ cắt tờ giấy và chỉ giữ lại 1 hình chữ nhật có tọa độ góc trái trên là (x, y) và tọa độ góc phải dưới là (u, v) . Bé muốn biết trong hình chữ nhật được giữ lại có bao nhiêu ô màu trắng. Vì tọa độ có thể rất lớn nên các bạn hãy lập trình để tìm kết quả giúp bé nhé.

Hình bên dưới mô tả cho góc phần tư thứ nhất khi được tô và hình chữ nhật cắt ra có tọa độ góc trái trên là $(4,8)$ và tọa độ góc phải dưới là $(9,1)$. Khi đó bé Cam sẽ nhận được 13 ô trắng.



Các bạn sẽ nhận được một bảng dữ liệu đầu vào bên dưới gồm 15 dòng tương ứng với 15 câu hỏi, mỗi câu hỏi sẽ gồm 4 số x, y, u, v nguyên dương. Bạn cần tạo file **HCN.TXT** gồm 15 dòng tương ứng với 15 câu hỏi, mỗi dòng ghi một số nguyên duy nhất là số ô trắng trong vùng (x, y, u, v) .

Test	X	Y	U	V
1	0	6	6	0
2	11	7	9	2
3	17	17	27	7

4	5	123	100	122
5	123	123	321	111
6	17	2017	2017	17
7	45	6789	9876	54
8	123	456789	456789	123
9	0	1000000	1000000	0
10	99	999999	888888	88
11	111	223344	556677	99
12	135	24680	97531	8642
13	2017	712017	172017	2017
14	987	654321	123456	789
15	500000	500000	1000000	0

Bài 3: ĐOÁN SỐ

Hôm nay lớp Cam có một trò chơi trong giờ Toán. Cô giáo nghĩ trong đầu một số nguyên X trong khoảng từ 1 đến 10^6 . Các bạn học sinh cần tìm được số bí mật X với ít nhất các câu hỏi có dạng: “Ước chung lớn nhất của số X với số Y là bao nhiêu?” Bạn nào dùng càng ít câu hỏi thì điểm càng cao.

Các bạn hãy giúp Cam viết một chương trình `doanso.*`, sử dụng các hàm/thủ tục trong thư viện `uclnlib.h` hoặc `uclnlib.pp` để tìm ra số bí mật X mà hết ít câu hỏi nhất.

Chương trình `doanso.*` của bạn phải khai báo sử dụng thư viện `uclnlib.h` hoặc `uclnlib.pp` theo cú pháp:

`#include<uclnlib.h>` hoặc `uses uclnlib.pp`

Các hàm/ thủ tục trong thư viện được định nghĩa như sau:

`int ucln(long long y)` hoặc `function ucln(y: int64): longint;`

Hàm này cho phép nhập vào một số nguyên dương $1 \leq Y \leq 10^{18}$ và trả về một số nguyên dương là giá trị ước chung lớn nhất của Y và số bí mật X .

`void traloi(int x)` hoặc `procedure traloi(x: longint);`

Thủ tục `traloi()` được dùng để trả về kết quả - là số X mà bạn xác định được.

Chương trình bắt buộc phải gọi thủ tục `traloi()` duy nhất một lần, nếu không sẽ bị 0 điểm. Thủ tục này khi được gọi sẽ tự động thoát chương trình.

Chương trình của các bạn không được chạy quá 1s và không được hỏi quá 1000000 lần. Nếu chạy quá 1s hoặc quá 1000000 lần thì ban giám khảo sẽ ngắt chương trình và ghi nhận điểm 0 với test đang chạy.

Số điểm cho mỗi test sẽ giảm dần khi số lần gọi hàm `ucln()` tăng lên.

Ví dụ: Số X cần tìm là 300000

Các hàm/thủ tục được gọi	Giá trị trả về	Giải thích
<code>ucln(100000)</code>	100000	Hỏi Ước chung lớn nhất của X với 100000. Khi kết quả trả về là 100000 ta biết rằng các số đó là bội của 100000. Trong khoảng $[1..10^6]$ có 10 số thỏa mãn dạng $100000 \times k$ với $k \in [1..10]$
<code>ucln(200000)</code>	100000	Ta loại trừ được 200000, 400000, 600000, 800000, 1000000.
<code>ucln(500000)</code>	100000	Ta loại trừ được số 500000.
<code>ucln(900000)</code>	300000	Vì chỉ còn 100000, 300000, 700000 và 900000 nên sau câu hỏi này ta đã tìm ra được đáp án là 300000. Nếu $X=100000$ thì hàm phải trả về 100000. Nếu $X=700000$ thì hàm phải trả về 100000. Nếu $X=900000$ thì hàm phải trả về 900000.
<code>tralo(300000)</code>		Như vậy bạn đã trả lời đúng với số lần hỏi là 4. Chương trình khi gọi <code>tralo(300000)</code> sẽ tự động đưa đáp án đồng thời thoát khỏi chương trình chạy của bạn.

Ghi chú: Trên máy làm bài của bạn đã được cung cấp 2 file `uclnlib.h` và `vidu.cpp` hoặc `uclnlib.pp` và `vidu.pas`

Bạn có thể tham khảo cách sử dụng thư viện và chương trình con trong file `vidu.cpp` hoặc `vidu.pas`.

Các bạn có thể sửa hằng số trong thư viện để chạy thử với dữ liệu khác.