



**Tổng quan về các bài trong đề thi:**

Bài	Tên bài	Tên file chương trình	Dữ liệu nhập từ:	Kết quả in ra:	Điểm
1	Điểm nằm trong	Inrectangle.cpp	Bàn phím	Màn hình	2.5
2	Chuỗi k -string	kstring.cpp	Bàn phím	Màn hình	2.5
3	Số gần nguyên tố	nearprime.cpp	Bàn phím	Màn hình	2
4	Vị trí cân bằng	equiposition.cpp	Equiposition.inp	Equiposition.out	1.5
5	Trạm thu sóng BTS	BTS1.cpp	BTS1.INP	BTS1.OUT	1.5

**Viết chương trình giải các bài toán sau:**

**Bài 1 (2,5 điểm): Điểm nằm trong (inrectangle.cpp)**

Bài toán: “Trên mặt phẳng tọa độ Oxy cho tứ giác lồi ABCD (các điểm A, B, C, D liệt kê đúng thứ tự theo cùng hoặc ngược chiều kim đồng hồ khi đi dọc biên của tứ giác) và hai điểm M, N. Hỏi rằng trong số hai điểm M, N có bao nhiêu điểm nằm bên trong tứ giác ABCD?”

Để giải bài toán trên Mai đã lấy bút chì vẽ tọa độ các điểm A, B, C, D, M, N trên tờ giấy và nhanh chóng nhận được câu trả lời. Thật không may, cậu em trai của Mai lấy tẩy xóa mất tọa độ của hai điểm B, D. Cho nên hôm sau Mai không thể nào tìm ra câu trả lời của mình hôm qua. Mai cầu cứu đến Hương - bạn thân học cùng với nhau từ nhỏ và rất giỏi toán. Tất nhiên với các dữ kiện trên Hương cũng chịu! Không chịu bó tay, Hương hỏi Mai: “Cậu còn nhớ thêm dữ kiện nào không?”. Sau một hồi suy nghĩ Mai nhớ ra “Tứ giác ABCD có dạng hình chữ nhật có các cạnh song song với một trong hai trục tọa độ!”. “Thế thì bài toán rất đơn giản!” Hương nói và nhanh chóng tìm ra câu trả lời.

Không tin tưởng lắm vào câu trả lời của Hương, Mai cầu cứu đến các bạn tham gia kỳ thi năng khiếu để nhờ viết một chương trình nhỏ kiểm tra xem Hương trả lời có đúng không?.

**Yêu cầu:** Biết tọa độ bốn điểm A, C, M, N hãy đếm xem trong số hai điểm M, N có bao nhiêu điểm nằm hẳn bên trong tứ giác ABCD (không tính trên biên).

**Dữ liệu:** Nhập từ bàn phím 8 số nguyên có trị tuyệt đối không vượt quá  $10^9$ ,  $x_A, y_A, x_C, y_C, x_M, y_M, x_N, y_N$  với  $(x_A, y_A)$  là tọa độ của điểm A,  $(x_C, y_C)$  là tọa độ của điểm C,  $(x_M, y_M)$  là tọa độ của điểm M,  $(x_N, y_N)$  là tọa độ của điểm N

**Kết quả:** In ra màn hình một số nguyên là câu trả lời: In số 2 nếu cả hai điểm M, N nằm hẳn bên trong (không nằm trên biên) tứ giác ABCD; in ra 1 nếu chỉ có điểm M hoặc N nằm



hắn bên trong (không nằm trên biên) tứ giác ABCD; in ra 0 nếu cả hai điểm M, N nằm bên ngoài tứ giác ABCD hoặc nằm trên biên của tứ giác này.

Ví dụ:

Dữ liệu nhập vào	Kết quả in ra
0 0	2
3 5	
1 1	
2 2	

**Bài 2. (2.5 điểm) chuỗi k -string (kstring.cpp)**

Một chuỗi được gọi là k-string là chuỗi ký tự có dạng lặp lại k lần một chuỗi nào đó. Ví dụ chuỗi "hoahoa" được gọi là chuỗi 2-string vì nó là kết quả của việc viết lặp lại 2 lần chuỗi "hoa". Chuỗi "abababab" vừa là chuỗi 2-string vừa là chuỗi 4-string, còn chuỗi "abc" chỉ là chuỗi 1-string. Tất nhiên một chuỗi bất kỳ luôn là chuỗi 1-string.

Yêu cầu: Cho số nguyên dương k và chuỗi ký tự s, hãy tìm cách sắp xếp lại thứ tự các ký tự trong s để chuỗi kết quả là một chuỗi k-string. Nếu không thể thực hiện được thì in ra số -1.

Dữ liệu vào:

- + Dòng đầu tiên chứa số nguyên k
- + Dòng thứ hai chứa chuỗi ký tự s khác rỗng và chỉ gồm các chữ cái tiếng anh in thường

Kết quả : Nếu có nhiều phương án thỏa mãn yêu cầu bài toán thì chỉ cần in ra một đáp án bất kỳ. Ngược lại, nếu không thực hiện được in ra số -1

Ví dụ:

Kstring.inp	Kstring.out
2 aabb	abab
2 abaa	-1

Ràng buộc:  $0 < k \leq 1000$ , chuỗi s có độ dài không quá 10000.

Có 60% số điểm ứng với số test ứng có chuỗi s độ dài không quá 200

**Bài 3 (2,0 điểm): Số gần nguyên tố (nearprime.cpp)**

Bộ phim mà Hương và Mai xem nói về cuộc đời nhà toán học Alan Turing - người đã chế tác thành công máy giải mã mà nhờ đó quân đội Anh đã phát hiện sớm nhiều chiến dịch quân sự của phát xít Đức trong chiến tranh thế giới lần 2 (1939-1945). Một trong những nền tảng quan trọng của việc giải mã là kiểm tra xem một số nguyên có phải là số nguyên tố. Tất nhiên Mai biết số nguyên tố là các số nguyên lớn hơn 1 và chỉ có hai ước dương (1 và chính nó). Ngồi cạnh "cây toán" của lớp Mai muốn mở rộng khái niệm này và hỏi Hương: "Những



số chỉ có 3 ước dương khác nhau thì như thế nào?”. “Thật tuyệt!” - Hương nói - “các số này là bình phương của một số nguyên tố, những số đầu tiên là 4, 9, 25, 49, 121, .... Tớ gọi đó là số gần nguyên tố!”.

Sẵn tờ vé xem phim có dãy số là seri của vé (có thể coi như là một số nguyên dương  $n$ ) Mai đề nghị Hương tìm một số “gần nguyên tố” nhỏ nhất lớn hơn hoặc bằng  $n$ . Tất nhiên cô bạn gái của chúng ta nhận được câu trả lời “ngay và luôn!” từ cậu bạn thân của mình. Còn bạn - những lập trình viên tương lai hiển nhiên không cần phải tính. Chúng ta chỉ cần viết một chương trình và máy tính tự nó cho câu trả lời!

**Yêu cầu:** Cho biết số nguyên dương  $n$ . Hãy tìm số gần nguyên tố nhỏ nhất lớn hơn hoặc bằng  $n$ .

**Dữ liệu:** Nhập từ bàn phím duy nhất số nguyên dương  $n$  ( $n \leq 10^{18}$ )

**Kết quả:** In ra màn hình số nguyên A là số gần nguyên tố nhỏ nhất lớn hơn hoặc bằng  $n$

**Ví dụ:**

Dữ liệu nhập vào	Kết quả in ra
20	25
3	4

**Ghi chú:** Bài được chấm trên 8 test, mỗi test 0,25 điểm. Trong đó

- 4 test có giá trị  $n \leq 100$
- 2 test tiếp theo có  $n \leq 10^6$
- 2 test còn lại không có ràng buộc bổ sung

#### Bài 4. (1.5 Điểm) Vị trí cân bằng (equiposition.cpp)

Cho một dãy A gồm N số nguyên được đánh số từ 1 đến N. Hãy tìm trong dãy A vị trí cân bằng đầu tiên tính từ trái sang phải. Vị trí  $i$  được gọi là cân bằng trong dãy nếu tổng các phần tử đứng bên trái và tổng các phần tử đứng bên phải vị trí  $i$  là bằng nhau.

Dữ liệu vào: File văn bản Equipostion.inp gồm có:

Dòng đầu tiên chứa số nguyên T biểu thị số bộ dữ liệu cần kiểm tra. Trong đó mỗi bộ dữ liệu gồm:

Dòng đầu tiên chứa số nguyên  $n$

Dòng thứ 2 chứa  $n$  số nguyên cách nhau 1 dấu cách biểu thị các phần tử của dãy A

Kết quả : File văn bản Equipostion.out

ứng với mỗi bộ dữ liệu đầu vào, in ra một dòng chứa vị trí cân bằng tìm được, nếu không có vị trí nào thỏa mãn thì in ra số -1

Ví dụ:

Equipostion.inp	Equipostion.out
2	1
1	3



1	
5	
1 3 5 2 2	

Ràng buộc:  $1 \leq T \leq 100, 1 \leq n \leq 10^5, 0 \leq a_i \leq 10^8$ .

**Bài 5. (1.5 điểm) Trạm thu sóng BTS (BTS1.cpp)**

Vịnh Hạ Long là một trong những thắng cảnh thiên nhiên nổi tiếng không chỉ ở Việt Nam mà còn trên Toàn thế giới. Trên Vịnh có rất nhiều hòn đảo đẹp là nơi tham quan của khách du lịch mỗi khi đến thăm vịnh. Để phục vụ nhu cầu của các khách tham quan, công ty viễn thông VNPT quyết định đặt một trạm thu sóng điện thoại (BTS) sao cho mọi khách tham quan trên các đảo có thể sử dụng điện thoại thông minh để liên lạc cũng như ghi nhận những cảm xúc của mình trên các mạng xã hội.

Để đơn giản, ta có thể coi bờ biển là đường thẳng trùng với trục Ox trên mặt phẳng tọa độ Đề-các. Mỗi hòn đảo như là một điểm có tung độ dương trên mặt phẳng này. Có tất cả  $n$  hòn đảo như vậy. Trạm BTS cần xây dựng là một điểm nằm ngay trên bờ biển (trục hoành) và vùng phủ sóng của trạm là hình tròn bán kính  $R$  với tâm là điểm đặt trạm. Tất cả các hòn đảo nằm trong hình tròn này (kể cả trên biên) đều bắt được sóng. Tất nhiên, để giảm chi phí đầu tư thì bán kính phủ sóng  $R$  càng nhỏ càng tốt

*Yêu cầu:* Hãy tìm vị trí đặt trạm BTS sao cho bán kính phủ sóng tối thiểu để phủ sóng toàn bộ số đảo trên vịnh là nhỏ nhất?. Để đơn giản, bạn chỉ cần tính giá trị nhỏ nhất này.

**Input:** File văn bản BTS1.INP gồm có:

- Dòng 1: Ghi số nguyên dương  $n$  ( $1 \leq n \leq 10^5$ )
- Dòng 2... $n + 1$ : Dòng  $i + 1$  ghi hai số nguyên  $x_i, y_i$  ( $|x_i| \leq 10^9, 0 < y_i \leq 10^9$ ) là tọa độ của hòn đảo thứ  $i$ .

**Output:** File văn bản BTS1.OUT

Ghi một số thực  $R$  duy nhất là bán kính phủ sóng tối thiểu tìm được.  $R$  được ghi với 2 chữ số sau phần thập phân.

**Example:**

BTS1.INP	BTS1.OUT
2	2.06
1 2	
2 2	