

Bài 13: Mật mã số nguyên tố

Peter muốn tạo ra một số số nguyên tố cho hệ thống mật mã của mình nhưng không lưu các số này vì sợ bị lộ. Vì vậy Peter chỉ lưu hai số nguyên A và B với ý đồ là mật mã của mình là các số nguyên tố trong đoạn từ A tới B. Bạn biết A và B, tìm dãy số nguyên tố trong mật mã của Peter. **Input.** Tập PRIPASS.INP có cấu trúc:

Dòng đầu tiên ghi số t ($t \leq 10$) là số test. t dòng tiếp theo, mỗi dòng ghi hai số A và B ($1 \leq A \leq B \leq 10^5$, $B-A \leq 10$)

Output. Tập PRIPASS.OUT với mỗi test trong tập Input, ghi ra dãy số nguyên tố của Peter từ bé đến lớn, mỗi số trên một dòng. Ngăn cách giữa các test ghi một dòng trống.

Ví dụ:

PRIPASS.INP	PRIPASS.OUT
2	3
3 5	5
1 10	2
	3
	5
	7

Bài 14: Số siêu không nguyên tố

Tim định nghĩa một số nguyên dương là số siêu không nguyên tố nếu xem số đó là một chuỗi các kí tự chữ số thì không có dãy con (theo thứ tự và không cần liên tục) nào của nó lập thành một số nguyên tố. Bạn được cho hai số A và B, hãy giúp Tim tìm các số siêu không nguyên tố thuộc đoạn từ A tới B.

Input. Tập NSPRIME.INP có cấu trúc:

Dòng đầu tiên chứa số t là số lượng bộ test ($1 \leq t \leq 10^5$), t dòng tiếp theo mỗi dòng ghi hai số A và B ($1 \leq A, B \leq 10^5$).

Output. Tập NSPRIME.OUT gồm t dòng, mỗi dòng ghi một số là số lượng các số siêu không nguyên tố tương ứng thuộc đoạn A, B cho trong tập Input.

Ví dụ:

NSPRIME.INP	NSPRIME.OUT
2	2
4 7	4
2 9	

Bài 15. Đoạn K_Nguyên tố

Một đoạn $[L..R]$ được gọi là một đoạn K_Nguyên tố nếu có ít nhất K số nguyên tố thuộc đoạn đó. Cho một số nguyên dương N và một số nguyên dương K. Hỏi có bao nhiêu đoạn con của đoạn $[2..N]$ là đoạn K_Nguyên tố?

Input: cho trong tệp **KPRIMESUB.INP** có cấu trúc như sau: Dòng đầu tiên chứa số T là số test.

T dòng tiếp theo, mỗi dòng chứa hai số nguyên dương N và K cách nhau một dấu cách.

Output: ghi ra tệp **KPRIMESUB.OUT** gồm T dòng, mỗi dòng tương ứng là câu trả lời với bộ dữ liệu vào cho trong tệp input. *Ràng buộc:* $1 \leq T \leq 10^2$, $2 \leq N \leq 10^5$, $0 \leq K \leq 10^4$ Ví dụ:

KPRIMESUB.INP	KPRIMESUB.OUT
3	4
5 2	9
5 1	8
9 3	

Bài 16. Biến đổi số nguyên tố

Cho hai số nguyên tố khác nhau, mỗi số có bốn chữ số. Người ta cho rằng hoàn toàn có thể biến đổi từ số này thành số kia sau một số bước theo quy tắc: Tại mỗi bước ta chỉ thay đổi một chữ số trong số trước đó sao cho số tạo được trong mỗi bước đều là một số nguyên tố có bốn chữ số.

Bài toán đặt ra là với một cặp số nguyên tố đầu vào, cần ít nhất bao nhiêu bước biến đổi theo quy tắc trên để biến số này thành số kia.

Ví dụ hai số 1033 và 8179 thì cần ít nhất 6 bước biến đổi như sau:

1033

1733

3733

3739

3779

8779

8179

Input: cho trong tệp **PRIMECHANGE.INP** có cấu trúc như sau: Dòng đầu tiên chứa số T là số test. T dòng tiếp theo, mỗi dòng chứa hai số nguyên tố có 4 chữ số là M và N cách nhau một dấu cách.

Output: ghi ra tệp **PRIMECHANGE.OUT** gồm T dòng, mỗi dòng tương ứng là số bước ít nhất để biến đổi M thành N cho trong tệp input.

Ràng buộc: $1 \leq T \leq 10$

Ví dụ:

PRIMECHANGE.INP	PRIMECHANGE.OUT
-----------------	-----------------

3	6
1033 8179	7
1373 8017	0
1033 1033	