

Thời gian làm bài: 180 phút
(không kể thời gian giao đề)

TỔNG QUAN ĐỀ THI

TT	Tên bài	Tên chương trình	Dữ liệu vào	Dữ liệu ra	Điểm	Thời gian
1	Addplus	Addplus.*	Addplus.inp	Addplus.out	100	1s
2	Shipper	Shipper.*	Shipper.inp	Shipper.out	100	1s
3	Stable Complex	StableComplex.*	StableComplex.inp	StableComplex.out	100	2s

(Trong đó * là PAS nếu sử dụng Pascal hoặc CPP nếu sử dụng C++ hoặc PY nếu sử dụng Python)

Hãy lập trình giải các bài toán sau:

Bài 1. Addplus

Cho 1 chuỗi s có độ dài n và chỉ gồm các chữ số từ '1' đến '9'.

Yêu cầu: Hãy thêm k dấu cộng chèn vào giữa các chữ số trên sao cho:

- Xâu được tạo ra là một biểu thức hợp lệ. (Ví dụ: "2 + 3 + 5" là biểu thức hợp lệ, nhưng +1, 1 + +2, 10+ thì không).
- Kết quả tạo ra sau khi thêm k dấu cộng là nhỏ nhất có thể.

Dữ liệu vào cho trong tệp: Addplus.inp

- Dòng 1: Gồm số nguyên T là số lượng test cases.
- Với mỗi test case:
 - Dòng thứ nhất: Gồm 2 số n ($2 \leq n \leq 2 \times 10^5$) và k ($1 \leq k \leq 6, k < n$) là độ dài chuỗi s và số lượng dấu cộng phải chèn.
 - Dòng thứ hai: Gồm chuỗi s có độ dài n và chỉ gồm các ký tự từ '1' đến '9'.

Dữ liệu đảm bảo tổng n trong tất cả test cases không vượt quá 2×10^5 .

Kết quả đưa ra tệp Addplus.out mỗi test case, in ra kết quả trên một dòng.

Ví dụ:

addplus.inp	addplus.out
3	2469
8 1	20
12341235	28
3 1	
911	
7 6	
1234567	

Chú ý: Trong test ví dụ:

- Test case 1 : Biểu thức tối ưu là: $1234 + 1235 = 2469$.

- Test case 2 : Biểu thức tối ưu là: $9 + 11 = 20$.
- Test case 3 : Biểu thức tối ưu là: $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$.

Giới hạn:

- Subtask 1 (25% số điểm) : $T = 1, n \leq 18$.
- Subtask 2 (20% số điểm) : $k = 1$.
- Subtask 3 (20% số điểm) : $k = 2$.
- Subtask 4 (35% số điểm) : Không có ràng buộc gì thêm.

Bài 2. Shipper

Thanh là một shipper chuyên nghiệp. Vào cuối ngày, Thanh còn n đơn hàng cần xử lý, được đánh số từ 1 đến n . Mỗi đơn hàng i có thời gian vận chuyển là t_i phút và mang lại tiền công là a_i đồng. Tuy nhiên, Thanh chỉ còn đúng T phút trước khi bưu cục đóng cửa và ngừng nhận xác nhận hoàn thành đơn hàng.

Để tối ưu lộ trình, Thanh có thể phân loại mỗi đơn hàng vào một trong hai nhóm: Giao Hỏa Tốc hoặc Giao Tiết Kiệm.

- Quy tắc ưu tiên: Thanh bắt buộc phải giao tất cả các đơn Hỏa Tốc trước, sau đó mới đến các đơn Tiết Kiệm.
- Thứ tự trong nhóm: Vì các đơn hàng đã được sắp xếp theo tuyến đường tối ưu trên bản đồ, nên trong cùng một nhóm, Thanh luôn giao theo thứ tự chỉ số từ nhỏ đến lớn.
- Điều kiện nhận công: Một đơn hàng i chỉ được tính là hoàn thành (và Thanh nhận được tiền công a_i) nếu tổng thời gian từ lúc bắt đầu ca cho đến khi giao xong đơn đó không vượt quá T phút.

Vì mỗi đơn hàng đều có thể được xếp vào nhóm "Hỏa Tốc" hoặc "Tiết Kiệm", nên có tổng cộng 2^n phương án phân loại khác nhau.

Yêu cầu: Bạn hãy giúp Thanh tính tổng số tiền công nhận được trên tất cả 2^n phương án phân loại có thể xảy ra. Vì con số này có thể rất lớn, hãy xuất ra kết quả sau khi chia lấy dư cho $10^9 + 7$.

Dữ liệu vào cho trong tệp Shipper.inp

- Dòng đầu tiên gồm hai số nguyên n và T ($1 \leq n \leq 200, 1 \leq T \leq 3 \times 10^5$).
- Dòng thứ hai gồm n số nguyên $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 3 \times 10^5$) — tiền công của từng đơn hàng.
- Dòng thứ ba gồm n số nguyên $t_1, t_2, \dots, t_n$ ($1 \leq t_i \leq T$) — thời gian xử lý từng đơn hàng.

Kết quả đưa ra tệp Shipper.out một số nguyên duy nhất là tổng tiền công thu được trên tất cả các cách phân loại, lấy modulo $10^9 + 7$.

Ví dụ:

shipper.inp	shipper.out
3 6 1 2 3 2 3 4	25

Giới hạn:

- Subtask 1 (25% số điểm) : $n \leq 20$.
- Subtask 2 (10% số điểm) : $t_i \geq T/2$ với mọi i .
- Subtask 3 (25% số điểm) : Tất cả đơn hàng đều có thời gian xử lý như nhau. ($t_1 = t_2 = t_3 = \dots = t_n$).
- Subtask 4 (40% số điểm) : Không có ràng buộc gì thêm.

Bài 3. Stable Complex

Trong một nghiên cứu về tương tác protein, Thanh đang mô phỏng các đại phân tử như những thành phần dao động có tần số riêng. Có n loại phân tử khác nhau, được đánh số từ 1 đến n . Phân tử thứ i có tần số dao động tự nhiên là i Hz.

Thanh muốn chọn một số phân tử để tạo thành một phức hợp protein. Với một nhóm phân tử, tần số cộng hưởng của phức hợp được xác định bằng bội chung nhỏ nhất (BCNN) của các tần số trong nhóm. Phức hợp đó được xem là ổn định nếu tần số cộng hưởng này là một bội số của tần số nền k .

Trong mô hình này, tần số nền k được biểu diễn thông qua các đơn vị cấu trúc chuẩn hóa, vì vậy mọi thừa số nguyên tố của k đều không vượt quá 10^6 .

Nhiệm vụ của bạn là đếm số tập con không rỗng của 1, 2, ..., n sao cho phức hợp tạo thành là ổn định.

Dữ liệu vào cho trong tệp StableComplex.inp

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 100$) là số bộ test.
- Mỗi bộ test gồm một dòng chứa hai số nguyên n và k ($1 \leq n, k \leq 10^{18}$)

Dữ liệu đảm bảo k không chia hết cho bất kỳ số nguyên tố nào lớn hơn 10^6 .

Kết quả đưa ra tệp StableComplex.OUT mỗi bộ test, in ra số lượng tập con thỏa mãn yêu cầu, lấy modulo $10^9 + 7$.

Ví dụ:

stablecomplex.inp	stablecomplex.out
4	8
4 4	24
6 12	184320
18 36	0
10 999983	

Giới hạn:

- Subtask 1 (20% số điểm) : $n \leq 20$.
- Subtask 2 (20% số điểm) : k là số nguyên tố.
- Subtask 3 (20% số điểm) : $n, k \leq 10^6$.
- Subtask 4 (20% số điểm) : $n, k \leq 10^{14}$.
- Subtask 5 (20% số điểm) : Không có ràng buộc gì thêm.

-HẾT-