Viết chương trình C++ đọc dữ liệu đầu vào về các chuyến giao hàng và thực hiện một số câu truy vấn

Input: dữ liệu gồm 3 phần:

Phần 1 gồm các chuỗi ngày liên , mỗi ngày trên một dòng theo định dạng YYYY-MM-DD (2022-01-23). Phần 1 được kết thúc bởi 1 dòng chứa \*

Phần 2 gồm thông tin về các chuyến giao hàng. Mỗi dòng mô tả thông tin về chuyến giao hàng:

<trip\_code> <customer\_code> <date> <time> qty

trong đó:

<trip\_code> là mã của chuyến giao hàng

<customer\_code> là mã khách hàng;

<date> <time> ngày giờ giao hàng của chuyến có mã <trip\_code> tới khách hàng có mã <customer\_code>, <date> có định dạng YYYY-MM-DD và <time> có định dạng hh:mm:ss

qty: số lượng gói hang

Mỗi chuyến hàng có thể được giao tới nhiều khách hàng. Số lượng chuyến có thể lên tới 10000 và mỗi chuyến có thể có 30 khách hàng.

Phần 2 kết thúc bởi dòng chứa \*\*\*

Phần 3 chứa câu truy vấn:

Truy vấn bao gồm nhiều dòng, mỗi dòng là thông tin về 1 loại câu truy vấn:

TOTAL\_QTY: trả về tổng số lượng gói hàng:

QTY\_CUSTOMER<customer\_code>: tổng số gói hàng đã giao cho khác hàng có mã customer\_code;

QTY\_MAX\_PERIOD <from\_date> <from\_time> <to\_date> <to\_time>: trong khoảng từ (<from\_date> <from\_time>) tới (<to\_date> <to\_time>) hãy tìm thời điểm (<date> <time>) mà tại đó tổng lượng giao hàng là lớn nhất, trả vê tổng lượng giao hàng.

Ví dụ, trong khoảng <from\_date> =2021-10-12 <from\_time> = 07:15:20 đến <to\_date> =2021-10-14 <to\_time> = 08:13:40, thời điểm lượng hàng giao lớn nhất là <date>=2021-10-13 <time> = 13:31:21 và tổng lượng hàng giao được lúc đó là 100, cần trả về 100

TOTAL\_TRIPS: số lượng chuyến giao hàng

TRAVEL\_TIME\_TRIP<trip\_code>: trả về thời gian làm việc của chuyến giao hàng có mã <trip\_code>

MAX\_CONFLICT\_TRIPS: tìm một tập các chuyến đi lấy trong cơ sở dữ liệu sao cho bất kỳ 2 chuyến đi nào thuộc tập cũng có khoảng thời gian làm việc bị trùng nhau và số lượng chuyến đi là lớn nhất. Trả về số lượng chuyến đi có trong tập tìm được

output: mỗi dòng chứa kết quả của 1 câu truy vấn

Ví dụ:

Input  
2020-01-02

2020-01-03

2020-01-04

2020-01-05

2020-01-06

\*

TR003 C002 2020-01-04 00:15:07 40

TR002 C001 2020-01-06 19:30:51 90

TR001 C001 2020-01-04 08:15:43 20

TR001 C003 2020-01-05 12:34:03 50

TR005 C003 2020-01-04 23:00:30 20

TR005 C001 2020-01-04 10:08:50 40

TR005 C001 2020-01-05 02:55:30 60

TR004 C003 2020-01-05 19:33:48 40

TR002 C001 2020-01-06 02:44:11 60

TR004 C001 2020-01-05 15:03:48 80

TR003 C003 2020-01-04 23:38:27 100

TR004 C001 2020-01-05 22:17:08 60

TR004 C002 2020-01-05 01:40:28 30

TR005 C002 2020-01-04 13:32:10 90

TR003 C002 2020-01-03 17:35:07 20

TR001 C001 2020-01-02 03:05:43 10

TR002 C003 2020-01-06 00:00:51 60

TR002 C002 2020-01-06 06:07:31 20

\*\*\*

TOTAL\_QTY

QTY\_CUSTOMER C002

QTY\_CUSTOMER C003

QTY\_MAX\_PERIOD 2020-01-02 01:04:17 2020-01-04 05:27:57

QTY\_MAX\_PERIOD 2020-01-02 20:21:46 2020-01-03 06:14:18

QTY\_MAX\_PERIOD 2020-01-02 02:58:24 2020-01-03 05:27:40

TOTAL\_TRIPS

TRAVEL\_TIME\_TRIP TR003

MAX\_CONFLICT\_TRIPS

\*\*\*

Output

890

200

170

40

0

10

5

108200

3

Một mạng truyền thông G bao gồm N máy chủ 1,...N được kết nối với nhau bởi M đường truyền (2 chiều). Giữa 2 máy chủ i và j có thể có 1 đường truyền hoặc không có. Trong trường hợp 2 máy chủ có đường truyền thì c(i,j) là chi phí thuê.

Công ty cần tính toán phương án thuê 1 số đường truyền nào đó của mạng G để kết nối liên thông một số máy chủ trong tập R (R là tập con của {1,2,..M} Tính toán đường truyền với chi phí thấp nhất.

INPUT :

Dòng 1 ghi 2 số nguyên dương N và M dòng 2 đến M. mỗi dòng ghi 3 số nguyên dương i,j,w: có đường truyền giữa i và j với chi phí w = c(i,j)

Dòng 2 ghi số nguyên K

Dòng 3 ghi số nguyên dương đôi một khác nhau là chỉ số của máy chủ trong R

OUTPUT: tổng chi phí thuê nhỏ nhất

Ví dụ:

INPUT:

5 8

1 2 1

1 3 2

1 5 5

2 4 7

2 5 6

3 4 30

3 5 4

4 5 8

4

1 2 3 4

Output:

10

Ta thực hiện thuê

=>dùng prim hoặc Kruskal tìm cây liên thông nhỏ nhất