

---

**Cotxes i distàncies****X23251\_ca**

---

Ens donen una llista de cotxes d'entrada. Per a cada cotxe ens donen el seu nom (un string), la màxima distància que poden recórrer amb el dipòsit ple (un natural) i el seu preu (un natural). Aquesta llista ens la donen ordenada de menys a més distància. Per exemple, ens poden donar la llista:

```
Corbette53 218 210513
Peugot307 415 33313
Ibiza2016 571 28216
Coupe2008 613 56198
Portofino 761 83690
```

Fixeu-vos que `Portofino` apareix en últim lloc perquè la seva distància (761) és la màxima. També ens donen una parella de valors enters  $[p_1, p_2]$ , que representa un interval de distàncies. Hem de considerar els cotxes que tenen una distància dins de l'interval  $[p_1, p_2]$ , calcular la suma dels seus preus, i escriure els noms d'aquests cotxes (en ordre invers al de la llista). Seguint amb l'exemple, suposeu que ens donen l'interval  $[312, 721]$ . Fixeu-vos que els cotxes `Peugot307`, `Ibiza2016` i `Coupe2008` són els que apareixen a la llista (en aquest ordre) amb una distància major o igual a 312 i menor o igual a 721. Llavors hauriem d'escriure la suma dels seus preus i els seus noms (en ordre invers):

```
117727 Coupe2008 Ibiza2016 Peugot307
```

Per a resoldre aquest exercici, és obligatori usar la següent declaració i implementar i usar convenientment les següents funcions:

```
struct Cotxe {
    string nom;
    int distancia;
    int preu;
};

typedef vector<Cotxe> Cotxes;

// Llegeix el nom, la distància i el preu d'un cotxe de l'entrada estandar,
// crea el cotxe amb aquestes dades i el retorna.
Cotxe llegirCotxe();

// Retorna cert sii el cotxe que és rep com a paràmetre té una distància
// dins de l'interval [p1,p2].
bool pertanyAInterval(const Cotxe &cotxe, int p1, int p2);
```

**Entrada**

La primera línia de l'entrada té un natural  $n$ , el nombre de cotxes. Després venen  $n$  línies, on cadascuna descriu un cotxe, amb el nom (un string de lletres majúscules i minúscules i dígitos), la distància que pot recórrer (un natural) i el preu (un natural). Després tenim

una nova línia amb un natural positiu  $m$ , el nombre de casos d'intervals. Després tenim  $m$  línies, on cadascuna té dos naturals  $p_1, p_2$ , que compleixen  $p_1 \leq p_2$  i descriuen un interval de distàncies  $[p_1, p_2]$ .

Sortida

Per a cadascun dels  $m$  casos  $[p_1, p_2]$ , s'ha d'escriure una línia. Aquesta línia conté, en primer lloc, la suma dels preus dels cotxes que tenen una distància dins de l'interval  $[p_1, p_2]$ . Després, la línia conté els noms d'aquests mateixos cotxes en ordre invers del de la llista.

Exemple d'entrada 1

5  
Corbette53 218 210513  
Peugot307 415 33313  
Ibiza2016 571 28216  
Coupe2008 613 56198  
Portofino 761 83690  
20  
312 721  
52 230  
52 73  
713 918  
799 905  
112 218  
112 117  
112 219  
761 1013  
760 1013  
762 1013  
218 415  
613 761  
219 414  
217 416  
612 762  
614 760  
454 699  
300 590  
73 1098

Exemple de sortida 1

117727 Coupe2008 Ibiza2016 Peugot307  
210513 Corbette53  
0  
83690 Portofino  
0  
210513 Corbette53  
0  
210513 Corbette53  
83690 Portofino  
83690 Portofino  
0  
243826 Peugot307 Corbette53  
139888 Portofino Coupe2008  
0  
243826 Peugot307 Corbette53  
139888 Portofino Coupe2008  
0  
84414 Coupe2008 Ibiza2016  
61529 Ibiza2016 Peugot307  
411930 Portofino Coupe2008 Ibiza2016 Peugot307 Corbette53

Exemple d'entrada 2

20  
a0 352 68  
a1 1545 20  
a2 1872 73  
a3 1981 17  
a4 3414 1  
a5 3418 50  
a6 4223 25  
a7 5155 46  
a8 5836 78  
a9 6246 95  
a10 6616 46  
a11 7060 74  
a12 7473 92  
a13 7569 88  
a14 8157 63  
a15 8795 66

a16 9249 38  
a17 9382 10  
a18 9489 12  
a19 9750 37  
20  
7646 9664  
4860 5085  
2650 7493  
3200 4090  
1229 1579  
4625 4734  
8204 11561  
8239 8278  
8623 12087  
9027 11562  
1087 2499  
3547 7209  
7870 9188  
9777 14146

5837 8111  
3649 5293  
872 2606  
67 4960  
2265 3133  
6165 8956

Exemple d’entrada 3

20  
a0 1 17  
a1 2 10  
a2 3 25  
a3 5 20  
a4 6 95  
a5 9 12  
a6 9 38  
a7 9 88  
a8 10 37  
a9 12 68  
a10 12 73  
a11 13 92  
a12 14 1  
a13 15 46  
a14 15 66  
a15 16 46  
a16 16 78  
a17 17 63  
a18 18 50  
a19 20 74  
10  
6 7  
20 27  
10 20  
20 30  
9 17  
5 11  
4 14  
19 22  
3 5  
7 17

Exemple de sortida 2

189 a18 a17 a16 a15 a14  
0  
507 a12 a11 a10 a9 a8 a7 a6 a5 a4  
51 a5 a4  
20 a1  
0  
163 a19 a18 a17 a16 a15  
0  
163 a19 a18 a17 a16 a15  
97 a19 a18 a17 a16  
110 a3 a2 a1  
364 a11 a10 a9 a8 a7 a6  
129 a15 a14  
0  
395 a13 a12 a11 a10 a9  
71 a7 a6  
110 a3 a2 a1  
254 a6 a5 a4 a3 a2 a1 a0  
0  
524 a15 a14 a13 a12 a11 a10 a9

Exemple de sortida 3

95 a4  
74 a19  
694 a19 a18 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8  
74 a19  
708 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5  
290 a8 a7 a6 a5 a4 a3  
524 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3  
74 a19  
45 a3 a2  
708 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5

## Observació

De cara a superar els jocs de proves públics i obtenir una bona nota (8 sobre 10 com a molt) n'hi ha prou amb que feu una implementació senzilla basada en cerques i recorreguts bàsics. Ara bé, els jocs de proves privats són grans ( $n$  i  $m$  són grans, tot i que el nombre de cotxes dins de cada interval demanat és petit en aquests jocs de proves privats). Per tant, si voleu aspirar a superar tots els jocs de proves privats i obtenir la màxima nota haureu d'implementar un esquema de cerca més eficient.

## Informació del problema

Autoria: FOPR

Generació: 2026-01-25T14:27:26.350Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>