

1)Într-un magazin sunt n obiecte; pentru fiecare se cunoaște greutatea G și valoarea V . Un hoț intră în magazin având un rucsac ce poate transporta o greutate maximă G_{Max} . El va fura anumite obiecte, sau porțiuni de obiecte, astfel încât suma greutăților obiectelor furate să nu depășească G_{Max} .

Să se stabilească câștigul maxim pe care îl poate obține hoțul. Câștigul este egal cu suma valorilor obiectelor furate. Câștigul adus de o fracțiune de obiect este direct proporțional cu greutatea fracțiunii.

Query:

4 30

10 60

5 50

12 60

20 140

Raspuns:

220

```

17 def cmp(e):
18     return e[2]
19     f = open("input.txt", "r")
20     vect = []
21     [n, G] = [int(x) for x in f.readline().split()]
22     for i in range(n):
23         vect.append([int(x) for x in f.readline().split()])
24         vect[i].append(vect[i][1] / vect[i][0])
25     vect.sort(reverse = True, key = cmp)
26     #print(vect)
27     rez = 0
28     for obj in vect:
29         if obj[0] > G:
30             rez = rez + G * obj[2]
31             G = 0
32         else:
33             rez = rez + obj[1]
34             G = G - obj[0]
35             if G == 0:
36                 break
37     print(rez)
38     f.close()

```

2) Se dă un șir cu n numere întregi. Determinați cel mai mare număr care poate fi scris ca produs de două elemente ale șirului.

Query:

7

-8 3 9 -1 -2 7 -10

Raspuns:

80

```

48     f = open("input.txt", "r")
49     n = int(f.readline())
50     vect = [int(x) for x in f.readline().split()]
51     vect.sort()
52     #print(vect)
53     rasp = max(vect[0] * vect[1], vect[n - 1] * vect[n - 2])
54     print(rasp)

```

3) La un festival sunt programate n spectacole.

Pentru fiecare se cunoaște momentul de început și momentul de sfârșit, exprimate prin numere naturale.

Un spectator dorește să urmărească cât mai multe spectacole în întregime.

Determinați numărul maxim de spectacole care pot fi urmărite, fără ca acestea să se suprapună.

Query:

10

5 14

14 17

8 13

13 15

15 17

3 6

4 7

6 9

11 14

10 11

Raspuns:

5

```
78 fin = open("data.in", "r")
79 fout = open("data.out", "w")
80
81 n = int(fin.readline())
82 intervale = []
83
84 for i in range(n):
85     [x,y] = [int(x) for x in fin.readline().split()]
86     intervale.append((x,y))
87     #print(x,y)
88
89     intervale.sort(key = lambda f: f[1])
90
91     print(intervale)
92
93     sol, last_end = 0, 0
94     for interval in intervale:
95
96         if interval[0] >= last_end:
97             last_end = interval[1]
98             sol = sol + 1
99     print(sol, file=fout)
100
101 fin.close()
102 fout.close()
```