

Descrierea generală a problemelor

Nr. crt.	Denumirea problemei	Restricția referitoare la volumul utilizat de memorie	Restricția referitoare la timpul de execuție, secunde	Punctajul alocat problemei
1.	Mașini	$\leq 1\text{Mb}$	$\leq 0,1$	100
2.	Bilete	$\leq 4\text{Mb}$	$\leq 0,1$	100
3.	Reducere	$\leq 1\text{Mb}$	$\leq 0,1$	100

Notă. În caz de egalitate de punctaj, prioritate se va da concurentului care a obținut punctajul respectiv primul.

Mașini

Ionel, elev în clasa a VI, mergând în drum spre școală și-a găsit o nouă distracție. El analizează numerele de mașini pe care le întâlnește în cale. Așa dar, Ionel știe: numere pare, numere impare, număr prim, număr compus, număr perfect, număr deficient, număr abundent și criteriile de divizibilitate cu 3,5,9,11. Astfel, Ionel citește numărul mașinii și încearcă să furnizeze cât mai multă informație despre acest număr.

Definiție: Notăm cu $\sigma(n)$ suma divizorilor numărului natural n , din care se exclude numărul însuși.

Astfel, avem definițiile: $\sigma(n) = \begin{cases} < n, & \text{număr deficient} \\ = n, & \text{număr perfect.} \\ > n, & \text{număr abundent} \end{cases}$

Sarcină: Elaborați un program care i-ar permite lui Ionel să furnizeze informația completă despre numerele de mașini întâlnite în cale.

Date de intrare: De pe prima linie a intrării standard se citește numărul natural n de trei cifre.

Date de ieșire: Pe linii separate a ieșirii standard se va afișa informația despre număr în următoarea ordine:

```
PAR
IMPAR
PRIM
COMPUS
PERFECT
DEFICIENT
ABUNDENT
SE DIVIDE CU 3
SE DIVIDE CU 5
SE DIVIDE CU 9
SE DIVIDE CU 11
```

Restricții: $100 \leq n \leq 999$. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `masini.pas`, `masini.c` sau `masini.cpp`.

Exemple:

Intrare

```
312
```

Ieșire

```
PAR
COMPUS
ABUNDENT
SE DIVIDE CU 3
```

```
363
```

```
IMPAR
COMPUS
DEFICIENT
SE DIVIDE CU 3
SE DIVIDE CU 11
```

Bilete

Ana foarte mult iubește să călătorească, respectiv căutarea biletelor de zbor este ceva foarte comun pentru ea. Obosită de căutarea și selectarea biletelor, Ana are nevoie de un program ce îi va ușura acest proces. Astfel, se propune crearea unui program ce va selecta dintr-o listă de bilete de avion, biletele ce corespund unei anumite clase și le va afișa în ordine descendentă după preț.

Sarcină. Elaborați un program care afișează la ecran prețurile biletelor în ordine descendentă ce corespund clasei dorite C .

Date de intrare. Intrarea standard conține în primul rând un număr întreg N ce reprezintă numărul de bilete, al doilea rând conține clasa C care reprezintă clasa dorită și este o literă majusculă din setul ASCII, apoi urmează N rânduri cu informația despre bilete, fiecare rând conține un număr întreg P_i prețul biletului și clasa biletului C_i o literă majusculă din setul ASCII separate prin spațiu.

Date de ieșire. Ieșirea standard va conține prețurile biletelor ce corespund clasei C afișate în ordine descrescătoare după preț toate într-un singur rând separate prin spațiu. Se știe că la ieșire va fi afișat cel puțin un bilet.

Restricții. $1 \leq N \leq 10^5$. $1 \leq P_i \leq 10^5$ Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea *bilete.pas*, *bilete.c* sau *bilete.cpp*.

Exemplu.

Intrare

```
6
B
120 A
200 B
400 B
170 F
40 A
15 B
```

Ieșire

```
400 200 15
```

Explicație.

Au fost selectate doar biletele de clasa B și au fost afișate în ordine descrescătoare după preț.

Reducere

“Black Friday” este un eveniment mult așteptat de toată lumea. Fiecare magazin își propune ofertele sale de reducere, dar pe Andrei l-a interesat oferta unui magazin de calculatoare. Acest magazin propune clienților săi următoarea ofertă: de la prețul unui produs p pot fi eliminate k cifre, astfel prețul final obținut după eliminarea a k cifre și este prețul ce trebuie achitat de către client.

Sarcină. Elaborați un program care determină cel mai mic număr posibil de obținut după eliminarea a k cifre din el.

Date de intrare. Intrarea standard conține un număr întreg p ce reprezintă prețul inițial a produsului și un număr întreg k ce reprezintă câte cifre trebuie eliminate din numărul p . Numerele sunt pe același rând separate prin spațiu. Se știe că numărul k nu va fi niciodată mai mare decât numărul de cifre din prețul p .

Date de ieșire. Ieșirea standard va conține un număr întreg care reprezintă cel mai mic preț posibil de obținut după eliminarea a k cifre. Prețul final nu trebuie să conțină zerouri ne semnificative la începutul numărului.

Restricții. $1 \leq p \leq 10^{18}$, $1 \leq k \leq 19$. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea *reducere.pas*, *reducere.c* sau *reducere.cpp*.

Exemplu.

Intrare

8900012 2

Ieșire

12

Explicație. Au fost eliminate cifrele 8 și 9, iar numărul final este 12. Răspunsul 00012 nu ar fi considerat răspuns corect.

Intrare

34020 3

Ieșire

0

Explicație. Au fost eliminate cifrele 3, 4 și 2.