`numere` - soluţia 1 –Programare dinamică. (Ilie Vieru) -100 puncte

- Dacă a are divizori primi mai mari decat 9 atunci scrie 0
- Altfel: $nr[1][d] = \begin{cases} 0, daca \quad b\%d <> 0, \\ 1, \quad in \quad rest. \end{cases}$

$$nr[i][j] = \sum_{\substack{1 \le p \le 9 \\ b \vdots p \\ j \vdots p}} nr[i-1][j/p]$$, unde $i \in \overline{2,a}$, iar $j \in \overline{1,b}$

Se afişează nr[a][b] (mod).

Problema se rafinează ajungând la complexitatea O(a*nrdiv(b)) înlocuind calculul matriceal cu cel cu vectori.

`numere` - soluţia 2 – backtracking (Stelian Ciurea) -100 puncte

Problema se poate rezolva şi prin metoda backtracking!

Pentru aceasta:
- se face descompunerea numărului **`b`** în factori primi (se observă că puterea maximă la care poate să apară un factor prim în această descompunere este **`13`** în cazul factorului prim **`2`**);
- în funcţie de această descompunere se reţine pentru fiecare cifră care este numărul maxim de câte ori poate să apară în numărul a (în vectorul **`f`**);
- se determină prin backtracking, succesiv, câte o configuraţie posibilă a numărului **`a`** astfel încât produsul ciferlor lui să fie **`b`**: astfel **`s[2]`** va reţine câte cifre de **`2`** apar în **`a`**, **`s[3]`** va reţine câte cifre de **`3`** apar în a etc;
- se determină numărul de cifre de **`1`** care apare într-o astfel de configuraţie;
- pentru o configuraţie posibilă se calculează toate numerele care au respectiva configuraţie a cifrelor, folosind formula aranjamentelor cu repetiţie:

$$A_b^{s[1],s[2],...,s[9]} = \frac{b!}{s[1]!s[2]!...s[9]!}$$

unde **`s[1]`** este numărul de cifre de **`1`** care apar în **`a`**, **`s[2]`** este numărul de cifre de **`2`** care apar în **`a`** etc.
- plecând de la observaţia evidentă că numărul de cifre de **`1`** este mult mai mare în comparaţie cu numărul celorlalte cifre (mai ales pentru valori mari ale lui **`b`**), se simplifică înainte de a începe calculul propriu-zis în formula anterioară **`b!`** cu **`s[1]!`**
- pentru a evita calcule repetate, înainte de apelarea subprogramului care implementează bactracking-ul descris anterior, se precalculează în vectorul **`fact`** factorialele numerelor de la **`2`** la **`13`** (**`13`** fiind factorialul maxim care apare la numitorul aranjamentelor cu repetiţii după simplificarea descrisă mai sus), precum şi puterile cifrelor de la **`1`** la **`13`** în matricea **`pow`**.