Cifru

- descrierea soluţiei –

I) Problema se poate reformula astfel: fie mulţimea $\{1,2,\ldots,n\}$ şi fie σ o permutare a acestei mulţimi care îndeplineşte condiţia:

$$\underbrace{\sigma(\sigma(\sigma(\ldots\sigma(}_{k \text{ ori}}x)\ldots))) = x \qquad \text{pentru orice } x \in \{1,2,\ldots,n\}$$

- Pentru început, vom da soluţia pentru un caz particular al problemei şi anume pentru K=3: să notăm cu f(n) numărul acestor permutări, şi calculăm în câte moduri 1 (primul element al mulţimii) poate să revină pe prima poziţie după K permutări. Există două posibilităţi şi anume:

1) $\sigma(1) = 1$ şi atunci evident că şi după K permutări successive 1 va rămâne pe prima poziţie; în acest caz, pentru celelalte n-1 elemente ale mulţimii vor exista f(n-1) permutări care îndeplinesc condiţia cerută;

2) elementul 1 formează un ciclu de lungime trei (evident cu încă alte două elemente din mulţime, fie acestea a şi b. Astfel $\sigma(1) = a$, $\sigma(a) = b$, $\sigma(b) = 1$ şi atunci aceste trei elemente revin pe poziţiile lor după trei permutări successive. În acest caz, cele două elemente a şi b putem să le alegem în $(n-1)(n-2) = A_{n-1}^{2}$ moduri, iar pentru restul de n-3 elemente ale mulţimii vom avea f(n-3) permutări care corespund cerinţei din enunţ.

Având în vedere aceste două posibilităţi putem scrie următoarea relaţie de recurenţă:

$f(n) = f(n-1) + (n-1)(n-2)f(n-3)$ şi
$f(n)=1$ pentru $n \le 1$

- Dacă K este număr prim, un raţionament asemănător ne conduce la următoarea relaţie de recurenţă:

$f(n) = f(n-1) + (n-1)(n-2)\ldots(n-k+1)f(n-k)$ şi
$f(n)=1$ pentru $n \le 1$

- Dacă K nu este număr prim, atunci raţionamentul anterior trebuie efectuat pentru fiecare divizor a lui K.
De exemplu, dacă K=6, atunci elementul 1 poate să revină pe prima poziţie după K permutări succesive dacă
- $\sigma(1) = 1$;
- elementul 1 formează un ciclu de lungime 2 (atunci revine pe prima poziţie după două permutări, după patru permutări şi în final după a şase permutări); celălalt element al acestui ciclu poate fi ales în A_{n-1}^{1} iar pentru mulţimea formată din celelalte n-2 elemente avem f(n-2) permutări;

- elementul 1 formează un ciclu de lungime 3 (atunci revine pe prima poziţie după trei permutări şi după a şase permutări);
- elementul 1 formează un ciclu de lungime 6 (atunci revine pe prima poziţie dupa şase permutări;

Obţinem astfel pentru cazul n=6, relaţia:

$$f(n) = f(n-1) + (n-1)f(n-2)+(n-1)(n-2)f(n-3) + (n-1)(n-2)\ldots(n-5)f(n-6)$$

sau

$$f(n) = f(n-1) + A_{n-1}^{1} f(n-2) + A_{n-1}^{2} f(n-3) + A_{n-1}^{5} f(n-6)$$

şi

$$f(n)=1 \quad \text{pentru } n \leq 1$$

Generalizâd, putem scrie recurenţa

$$f(n) = f(n-1) + \sum_{d=2}^{k} A_{n-1}^{d-1} f(n-d)$$ pentru toţi divizorii d ai lui K care sunt mai mici sau egali cu n.

Programul constă în implementarea acestei formule. Pentru a obţine punctaj maxim, trebuie făcute câteva optimizări cum ar fi:

- calculul A_{n-1}^{d2} unde d2 este un divizor a lui K trebuie făcut plecând de la valoarea A_{n-1}^{d1} calculată în prealabil, unde d1< d2 şi d1 divizor a lui K
- reţinerea valorilor calculate pentru funcţia f într-un vector, pentru a evita calcularea în mod repetat a acestora.

Fără aceste optimizări, un program care implementează soluţia descrisă primeşte 50 puncte.

II) O soluţie bazată pe backtracking, care generează toate permutările şi verifică pentru fiecare dacă îndeplineşte condiţia din enunţ este corectă, dar se încadrează în timpul maxim de rulare/test doar pentru valori mici ale lui n şi k. Pentru datele de test propuse, o astfel de soluţie primeşte 20 puncte;

III) Se pot lua 30 puncte cu o astfel de soluţie dacă se rulează programul în timpul concursului petru toate combinaţiile posibile sugerate de observaţia că

- pentru `30%` din teste `N,K < 13`

şi se reţin rezultatele într-o matrice de constante.

În subdirectorul solutii se află două surse de 100 puncte:

- stelian.cpp, care foloseşte o implementare recursivă a formulei;
- cosmin.pas, care foloseşte un algoritm interativ şi are, în consecinţă, o viteză de rulare sensibil mai bună.