

Permutări

Prof. Nistor Moț

1. Noțiuni introductive

Se numește **permutare** o funcție bijectivă $p:A \rightarrow A$.

De obicei se consideră $A=\{1,2,\dots,n\}$ și permutarea se scrie sub forma
$$\begin{pmatrix} 1 & 2 & 3 & \dots & n \\ p(1) & p(2) & p(3) & \dots & p(n) \end{pmatrix}$$
 dar toate rezultatele rămân valabile dacă se înlocuiește mulțimea $\{1,2,\dots,n\}$ cu orice mulțime cu n elemente (evident, în programare nu se lucrează cu mulțimi infinite).

De asemenea ordinea numerelor din primul rând este neesențială, de exemplu permutarea $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}$ poate fi scrisă și $\begin{pmatrix} 2 & 3 & 1 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$, în ambele cazuri $p(1)=2$, $p(2)=4$, etc.

Vom nota mulțimea permutărilor lui $\{1,2,\dots,n\}$ cu S_n .

Se numește **permutarea identică** (și se notează de obicei cu e) funcția identică a mulțimii A (1_A) adică permutarea p în care $p(1)=1$, $p(2)=2$, ..., $p(n)=n$.

Se numește **punct fix** al unei permutări $p \in S_n$ un număr $x \in \{1,2,\dots,n\}$ pentru care avem $p(x)=x$. Evident permutarea identică are n puncte fixe

Se numește **transpoziție** o permutare $p \in S_n$ cu $n-2$ puncte fixe. Dacă $p(i)=j$ și $p(j)=i$, restul valorilor fiind puncte fixe, transpoziția se mai notează și $(i\ j)$.

Operația de bază în mulțimea S_n este compunerea permutărilor, o vom nota cu o permutările fiind funcții este vorba despre compunerea funcțiilor. Se cunosc proprietățile :

compunerea este asociativă – adică $(poq)or=po(qor)$

compunerea nu este comutativă - adică în general $poq \neq qop$

permutarea identică este element neutru – adică $poe=p$ și $eop=p$

fiecare permutare are o permutare inversă astfel încât compunând cele două permutări obținem permutarea identică - $pop^{-1} = e$ și $p^{-1}op = e$. Inversa unei permutări se poate obține inversând cele două rânduri, apoi eventual reordonând tabloul în ordinea crescătoare a numerelor de pe primul rând. Pentru exemplul de mai sus

inversa este
$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

O transpoziție compusă cu ea însăși dă permutarea identică, deci este propria ei inversă.

Problema 1.

Calculați câte cifre are numărul de permutări pentru $n \leq 1000000000$.

Rezolvarea se poate face folosind proprietățile logaritmilor : $\lg n! = \lg 1 + \lg 2 + \dots + \lg n$, partea întreagă a logaritmului zecimal +1 ne va da numărul de cifre. Dar un “for” mergând de la 1 la 1000000000 depășește timpul normal pentru execuția unui program (dacă calculatorul adună 10.000.000 logaritmi pe secundă vor fi necesare 100 secunde !). Există o formulă celebră (formula lui Stirling) ce aproximează foarte bine (pentru numere mari) pe $n!$, pentru $n > 100$ numărul de cifre va fi dat fără eroare de această formulă, chiar și primele cifre sunt date corect. Formula este :

$$n! = \left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n} \quad (\text{în formulă apar două numere iraționale } e \approx 2,71 \text{ și } \pi \approx 3,14)$$

Cu această formulă logaritmul lui $n!$ se calculează imediat $n \cdot (\lg n - \lg e) + (\lg n + \lg 2\pi)/2$.

2. Inversiuni

Se numește **inversiune** a unei permutări $p \in S_n$ o pereche (i, j) cu $1 \leq i < j \leq n$ și $p(i) > p(j)$. Numărul inversiunilor unei permutări joacă un rol important în multe aplicații ale permutărilor.

Problema 2. (din lista lui Frâncu)

Se dau două numere naturale N și K , $1 \leq N \leq 45$, $0 \leq K \leq N \cdot (N-1)/2$. Sa se tipareasca numărul de permutări ale multimii $\{1, 2, \dots, N\}$ în care există K inversiuni.

INTRAREA: Fișierul `perm.in` conține o singură linie pe care se afla valorile lui N și K , despartite printr-un spațiu.

IESIREA: În fișierul `perm.out` se va afișa un singur număr (nimic altceva, svp) respectiv numărul de permutări cu K inversiuni.

EXAMPLE:

<code>perm.in</code>	<code>perm.out</code>
3 0	1
<code>perm.in</code>	<code>perm.out</code>
4 2	5

COMPLEXITATE RECOMANDATA: $O(N^2 \cdot C)$, unde C este numărul de cifre al răspunsului.

BREVIAR: PERMUTARI (pentru cei mici)

Permutările unei multimi sunt toate ordinele posibile în care putem așeza elementele acelei multimi. De exemplu, permutările multimii $\{1, 2, 3\}$ sunt: 123, 132, 213, 231, 312, 321

Se poate arăta că numărul permutărilor unei multimi cu N elemente este egal cu $N!$.

Dându-se o permutare P , numărul de inversiuni al ei este numărul de perechi (i, j) pentru care $i < j$ și $P[i] > P[j]$. De exemplu, pentru permutarea cu 5 elemente: $P = 52314$, perechile (i, j) în dezordine sunt:

- (1,2): 1 < 2 dar 5 > 2
- (1,3): 1 < 3 dar 5 > 3
- (1,4): 1 < 4 dar 5 > 1
- (1,5): 1 < 5 dar 5 > 4
- (2,4): 2 < 4 dar 2 > 1
- (3,4): 3 < 4 dar 3 > 1

Asadar permutarea în cauză are 6 inversiuni. Numărul minim de

inversiuni al unei permutări de N elemente este 0 (pentru permutarea $1\ 2\ 3\ \dots\ N-1\ N$), iar numărul maxim de inversiuni este $N*(N-1)/2$ (pentru permutarea $N\ N-1\ \dots\ 3\ 2\ 1$).

Cam atât ajunge, pentru început. Mai am în minte o problemă cu permutări.

Rezolvarea este tot pe lista lui Frâncu, se bazează pe o formulă de recurență.

În [1] pag. 51 există un exercițiu care dă câteva formule pentru calculul rezultatului cerut de problemă.

La concursul de programare Agora organizat de GInfo s-a dat problema următoare (existentă în [2], pag. 16) :

Problema 3.

Se dă un vector a care conține n ($2 \leq n \leq 10000$) elemente și se cere determinarea numărului de perechi (i, j) cu proprietatea că $i < j$ și $a[i] > a[j]$ (timp de execuție 1s !)

Problema e rezolvată și comentată în GInfo 11/5, an 2001.

3. Ordinul unei permutări

Se numește **ordinul** permutării $p \in S_n$ cel mai mic număr $n \in \mathbb{N}^*$ cu proprietatea că $p^n = e$ (prin p^n înțelegând $\text{popop} \dots \text{op}$, p compus cu el însuși de n ori)

Demonstrarea existenței unui astfel de număr :

Considerăm șirul $p, p^2, p^3, \dots, p^{100}, \dots$. Cum există doar un număr finit de permutări în S_n (adică $n!$) șirul scris trebuie să conțină permutări care se repetă de mai multe ori, fie $p^a = p^b$ ($a > b$). Atunci $p^{a-b} = e$, prin compunere cu p^{-b} (adică inversul lui p^b sau $(p^{-1})^b$).

Noțiunea e necesară în rezolvarea unor probleme de genul :

Problema 4.

Fie $p \in S_n$ o permutare și $k \in \mathbb{N}^*$. Determinați p^k . Fișierul de intrare PERM.IN conține pe prima linie numerele n și k , iar pe linia a doua cele n numere ce formează permutarea. În fișierul PERM.OUT se va scrie răspunsul sub forma unui șir de numere scrise pe o linie cu câte un spațiu între numere. $2 \leq n \leq 1000$, $1 \leq k \leq 1000000000$.

Pentru determinarea ordinului unei permutări se folosește descompunerea permutării în cicluri.

Să pornim de la un exemplu. Fie $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{pmatrix}$. Avem $p(1)=3$, $p^2(1) = p(p(1))=p(3)=2$, $p^3(1)=p(p(p(1)))=p(2)=1$, deci după 3 permutări 1 a ajuns pe poziția 1, parcurgând ciclul $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$. Să observăm că 2 și 3 urmează un drum analog : $2 \rightarrow 1 \rightarrow 3 \rightarrow 2$ și $3 \rightarrow 2 \rightarrow 1 \rightarrow 3$. Asemănător găsim ciclul $4 \rightarrow 5 \rightarrow 4$. Deducem că dacă 1, 2 și 3 ajung în pozițiile 1, 2, 3 la p^3 , p^6 , $p^9, \dots$ adică din 3 în 3 permutări iar 4 și 5 ajung pe pozițiile 4, respectiv 5, la p^2 , $p^4, \dots$ ordinul permutării va fi cmmmc al numerelor 2 și 3, adică 6. Să menționăm că ciclurile se scriu asemănător transpozițiilor, în cazul nostru $(1\ 3\ 2)$ respectiv $(4\ 5)$. Se numește lungimea unui ciclu numărul elementelor (distincte) ce alcătuiesc ciclul. Un punct fix formează un ciclu de lungime 1, o transpoziție, un ciclu de lungime 2.

Cum determinăm deci ordinul unei permutări ? Se pornește de la 1, parcurgând vectorul ce reprezintă permutarea și marcând elementele 1, $p[1]$, $p[p[1]]$, până ajungem la un element marcat (adică la 1), numărând numărul de pași în l1. Căutăm primul element nemarcat din vector și reluăm parcurgerea, efectuând l2 pași. Când nu mai avem elemente nemarcate calculăm cmmmc al numerelor l1, l2, ..., lk. Marcarea se poate face schimbând semnul numerelor din vector, iar cmmmc se poate calcula la

Problema a fost prezentată la o ședină de pregătire cu lotul de informatică anul trecut la Slatina

5. Probleme de numărare

Se pun uneori probleme în care se cere numărarea unor permutări cu diverse proprietăți. Exemplificăm cu două astfel de probleme

Problema 7.

Se da $n \leq 100$. Câte permutări p din S_n verifică $p \circ p = e$

Se va deduce relația de recurență $T(n) = T(n-1) + (n-1) \cdot T(n-2)$ astfel : Permutarea trebuie să conțină numai cicluri de lungime 1 sau 2. Dacă $p(n) = n$ cele $n-1$ numere din fața lui n pot fi așezate în $T(n-1)$ moduri pentru a obține o permutare cu proprietatea din enunț. Dacă $p(n) \neq n$, va exista $k < n$ astfel ca $p(n) = k$ și $p(k) = n$ (altfel n ar face parte dintr-un ciclu de lungime > 2). Numerele din $\{1, 2, 3, \dots, n\} - \{k, n\}$ pot fi așezate în $T(n-2)$ moduri și cum k poate fi oricare din cele $n-1$ numere avem $(n-1) \cdot T(n-2)$ permutări ce verifică relația cerută în care $p(n) \neq n$. Totalul va fi dat de suma celor doi termeni. Ca observație pentru obținerea răspunsului corect e necesară implementarea operațiilor cu numere mari, chiar dacă numărul cerut este o fracțiune foarte mică din totalul permutărilor (de exemplu o milionime), totuși pentru $n > 50$ rezultatul depășește numărul de cifre exacte ale tipurilor numerice cunoscute.

Problema 8.

Considerăm n persoane numerotate de la 1 la n ($n \leq 1000$) care iau masa zilnic, așezați la o masă circulară. Dorim să așezăm persoanele astfel încât în fiecare zi să aibă vecini diferiți.

a) Afișați numărul maxim de zile în care niciunul din vecinii vreunei persoane n-a mai fost vecin în zilele anterioare și dați un exemplu de astfel de așezare

b) Găsiți numărul maxim de zile dacă în fiecare zi trebuie să fie cel puțin un vecin diferit față de zilele precedente (adică numărul maxim de aranjări diferite)

Exemplu : pentru $n=5$ a) două zile : 1-2-3-4-5, 1-4-2-5-3

b) 6 zile

Rezolvarea prin generarea tuturor aranjamentelor este evident ineficientă.

6. Probleme cu inegalități

Se cunosc din celebrele culegeri de probleme de algebră (gen Năstăsescu & Niță) exerciții de genul :

Pentru ce permutare $p \in S_n$ suma $1/p(1) + 2/p(2) + \dots + n/p(n)$ este minimă / maximă?

Pentru ce permutare suma $|1-p(1)| + |2-p(2)| + \dots + |n-p(n)|$ este minimă / maximă

Se dau numerele $x_1 < x_2 < \dots < x_n$. Găsiți permutarea p pentru care suma $(x_1 - x_{p(1)})^2 + (x_2 - x_{p(2)})^2 + \dots + (x_n - x_{p(n)})^2$ este minimă/maximă.

La aceste întrebări răspunsul este de obicei permutarea identică pentru minim și permutarea $\begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \end{pmatrix}$ pentru maxim sau invers.

Un exemplu clasic :

Se dau n numere reale pozitive ($n \leq 10000$), $x_1, x_2, \dots, x_n$. Găsiți o permutare $p \in S_n$ pentru care suma $x_1 \cdot p(1) + x_2 \cdot p(2) + \dots + x_n \cdot p(n)$ este maximă.

Nu întotdeauna răspunsul la astfel de probleme este una din cele două permutări.

Problema 9.

Determinați permutarea p din S_n pentru care $|p_1 - p_2| + |p_2 - p_3| + \dots + |p_n - p_1|$ este maximă

Sol. pt n par avem $p(2k+1) = n/2 - k$, $p(2k+2) = n - k$ ($k=0, 1, \dots, n/2 - 1$)

pt. n impar $p(2k+1) = \lfloor n/2 \rfloor - k$, $p(2k+2) = n - k$ ($k=0, 1, \dots, \lfloor n/2 \rfloor + 1$)

sau **Problema 9'.**

Determinați permutarea p din S_n pentru care $p_1 p_2 + p_2 p_3 + \dots + p_n p_1$ este maximă

Sol. permutarea pentru maxim este $(1, 2, 4, 6, \dots, 7, 5, 3)$ iar pt. minim $(1, n, 2, n-2, \dots, 5, n-3, 3, n-1)$

Soluția de la prob.9 amintește de o foarte actuală problemă (dată la Olimpiada-online, etapa 15 oct – 14 nov. 2001) în care se cere numărul permutărilor în zig-zag.

Considerăm funcția $E: S_n \rightarrow \mathbb{N}$, $E(p) = 1 \cdot p(1) + 2 \cdot p(2) + \dots + n \cdot p(n)$.

Este funcția E injectivă sau surjectivă ?

- Injectivă nu este deoarece $E(p) = E(p^{-1})$

- E ia valoarea maximă pentru $p=e$ și valoarea minimă pentru $p = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \end{pmatrix}$

deci ia valori între $\min = n(n+1)(n+2)/6$ și $\max = n(n+1)(2n+1)/6$, așadar funcția nu e nici surjectivă. Dar dacă restrângem codomeniul la intervalul $[\min, \max]$ putem demonstra că E ia toate valorile din acest interval ? Pentru un $S \in \mathbb{N}^*$ putem găsi $p \in S_n$ astfel ca $E(p) = S$?

Acesta este enunțul foarte puțin modificat al unei probleme date la barajul pentru lotul internațional în 2001 la Bacău

Problema 10 – Potrivire (Baraj ONI 2001)

Se consideră două numere naturale nenule N și S .

Cerinta

Determinați numerele distincte $x_1, x_2, \dots, x_N$ aparținând mulțimii $\{1, 2, \dots, N\}$ astfel încât: $1 \cdot x_1 + 2 \cdot x_2 + \dots + N \cdot x_N = S$.

Date de intrare

Fișier de intrare: **POTRIV.IN**

Linia 1: $N \ S$

- două numere naturale nenule, separate printr-un spatiu, reprezentand numarul de numere, respectiv suma ce trebuie obtinuta.

Date de iesire

Fișier de ieșire: **POTRIV.OUT**

Linia 1: $x_1 \ x_2 \ \dots \ x_N$

- N numere naturale nenule, separate prin cate un spatiu, reprezentand solutia problemei. Daca nu exista solutie, pe aceasta linie se va scrie numarul 0.

Restricții

- $2 \leq N \leq 1000$
- $1 \leq S \leq 1000000000$ (un miliard)
- Dacă există mai multe soluții, în fișier se va scrie una singură.

Exemplu

POTRIV.IN

4 26

POTRIV.OUT

3 2 1 4

Explicație

$$1*3 + 2*2 + 3*1 + 4*4 = 26$$

Rezolvare (propusa de reprezentantul județului Brăila la baraj)

Dacă S nu este în intervalul $[\min, \max]$ nu există soluție. Altfel pornim cu permutarea identică, care dă suma maximă și facem transpoziții până ajungem la suma S . Fiecare transpoziție micșorează suma (notată anterior cu $E(p)$). Dacă schimbăm $p(i)$ cu $p(j)$, restul elementelor rămânând neschimbate, suma scade cu $(i-j)*(p(i)-p(j))$. De aici mai e un pas (dar nu banal) până la obținerea soluției.

7. Codul Gray

Considerăm graful în care nodurile sunt permutări din S_n iar între două permutări avem muchie dacă permutările diferă doar în două poziții adiacente (de ex. în S_5 avem muchie între (13524) și (13254), dar nu între (13524) și (12534)). Găsiți un ciclu hamiltonian în acest graf.

Cu alte cuvinte să generăm permutările într-o ordine asemănătoare codului Gray (chiar mai restrictiv), pornind de la permutarea identică și interschimbând de fiecare dată câte două valori alăturate. Faptul că e posibil acest lucru rezultă prin inducție: presupunând că am generat un astfel de șir al permutărilor din S_{n-1} , pentru generarea permutărilor din S_n trebuie doar să adăugăm pe n în mod convenabil.

Exemplificăm pentru $n=4$. Așezăm pe primul rând al unei matrice șirul de permutări generat pentru 3 (adică 123, 132, 312, 321, 231, 213) iar pe fiecare coloană intercalăm pe 4 în toate pozițiile pornind de la sfârșit:

1234	1324	3124	3214	2314	2134
1243	1342	3142	3241	2341	2143
1423	1432	3412	3421	2431	2413
4123	4132	4312	4321	4231	4213

Acum parcurgând matricea pe coloane alternativ (de sus în jos apoi de jos în sus etc) obținem șirul de permutări dorit.

O legătură mai strânsă cu codul Gray se obține numărând inversiunile fiecărei permutări. Se obține codul Gray pentru baza mixtă 1,2,3,4.

Algoritmul de generare de mai sus

P1: $c=(0,0,0,\dots,0)$; $d=(1,1,1,\dots,1)$ (vectorul c pentru inversiuni, d pt. direcție)

P2: vizitează permutarea $a_1a_2\dots a_n$

P3: $j=n$; $s=0$ (s este numărul indicilor $k>j$ pt. care $c_k=k-1$)

P4: $q=c_j+d_j$. Dacă $q<0$ mergi la P7, dacă $q=j$ mergi la P6

P5: schimbă $a_{j-c_j+s} \leftrightarrow a_{j-q+s}$; $c_j=q$; mergi la P2

P6: dacă $j=1$ stop altfel $s=s+1$

P7: $d_j=-d_j$; $j=j-1$; mergi la P4

Problema 11

Se dă o permutare din S_n . A câta este generată de procedeul anterior?

Mergând pe ideea de mai sus, cu graful având S_n ca mulțime de noduri, fie $Q=\{p_1, p_2, \dots, p_k\}$ o mulțime de permutări. Vom lega prin muchii fiecare permutare p cu cele k permutări $pop_1, pop_2, \dots, pop_k$. Mai putem găsi un ciclu hamiltonian? În general nu.

8. Probleme finale

Înceiem cu două probleme, una ușoară, legată tot de descompunerea unei permutări în cicluri și o problemă frumoasă, dar grea, dată la Olimpiada Internațională de Informatică din China în 2000.

Problema 12.

Rezolvați ecuația $x^3 = y$, x și y fiind permutări din S_n ($n \leq 1000$).

În fișierul de intrare "EC.IN" e scris pe prima linie numărul n , iar pe linia următoare numerele $y(1), y(2), \dots, y(n)$ cu câte un spațiu între ele. În fișierul de ieșire "EC.OUT" trebuie să scrieți permutarea x , numerele $x(1), x(2), \dots, x(n)$ fiind scrise pe o linie, cu spații între ele. Dacă sunt mai multe soluții, se va scrie una singură, dacă nu există soluții se va scrie numărul 0.

Problema se poate rezolva prin încercări, se obține un algoritm cubic, acceptabil pentru valori mai mici ale lui n . Pentru valori mari apelăm din nou la descompunerea

în cicluri. Fie $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 6 & 1 & 2 & 4 \end{pmatrix} = (1 \ 3 \ 6 \ 4) \bullet (2 \ 5)$. Avem $p^3 = (1 \ 4 \ 6 \ 3) \bullet (2 \ 5)$.

Se observă că un ciclu de lungime $l > 3$, l nedivizibil cu 3, compus de 3 ori cu el însuși dă tot un ciclu de lungime l . Afirmatia e valabilă și pentru cicluri de lungime 1 sau 2. Pentru ciclurile de lungime l divizibilă cu 3, prin compunerea de 3 ori se obțin 3 cicluri de lungime $l/3$. Reconstituirea, dacă e posibilă se face destul de ușor.

O varianta mai literara:

Cea de-a doua problemă finală diferă de cele de până aici nu numai prin gradul de dificultate și prin modul de obținere a datelor de intrare sau de ieșire.

Problema cere în fond identificarea termenului median dintr-o permutare $p \in S_n$ ($n < 1500$), adică $p^{-1}((n+1)/2)$. Singura problemă e că nu se dă permutarea ci un dispozitiv (o funcție) care ne spune pentru 3 numere $a, b, c \in \{1, 2, \dots, n\}$ care din numerele $p(a), p(b), p(c)$ este între celelalte două. Dispozitivul (funcția) poate fi apelat de max. 7777 ori. Enunțul complet este :

Problema 13. – Puterea mediană, IOI 2000

Un experiment implică N obiecte, etichetate de la 1 la N . Se știe că N este impar. Fiecărui obiect i se atașează o informație (numită putere) care este necunoscută și care este exprimată printr-un număr natural; puterile obiectelor sunt distincte. Fiecare putere P satisface relația $1 \leq P \leq N$. Obiectul de putere mediană este obiectul X cu proprietatea că numărul de obiecte cu puterea mai mică decât puterea lui X este egal cu numărul de obiecte cu putere mai mare decât puterea lui X . Scrieți un program care determină obiectul de putere mediană. Din nefericire, singurul mod de a compara puterile este bazat pe un dispozitiv care, pentru 3 obiecte distincte date, determină pe acela dintre ele care are puterea mediană.

Biblioteci

Este dată biblioteca *device* care conține 3 subprograme :

- GetN – trebuie apelat o singură dată, la început, nu are argumente și returnează valoarea N
- Med3 – trebuie apelat cu etichetele a 3 obiecte distincte ca argumente și returnează eticheta obiectului de putere mediană
- Answer – trebuie apelat o singură dată la sfârșit și are ca argument eticheta unui obiect; acest obiect ar trebui să fie cel care are puterea mediană; această operație este cea care încheie executarea programului.

Biblioteca *device* este folosită pentru a crea două fișiere text MEDIAN.OUT și MEDIAN.LOG. Prima linie a fișierului MEDIAN.OUT conține un întreg indicând eticheta obiectului transmis ca parametru în apelarea subprogramului Answer. A doua linie conține un număr întreg care indică numărul de apeluri ale subprogramului Med3 efectuate de programul dumneavoastră. Dialogul între program și bibliotecă este memorat în fișierul MEDIAN.LOG.

Programatorii în Pascal trebuie să include în codul sursă instrucțiunea `uses device;`

Programatorii în C/C++ trebuie să includă în codul sursă instrucțiunea `#include "device.h"`.

Trebuie creat proiectul MEDIAN.PRJ și apoi adăugate fișierele MEDIAN.C (MEDIAN.CPP) și DEVICE.OBJ în acest proiect.

Experimentare

Lucrul cu biblioteca se poate experimenta creând un fișier text DEVICE.IN. Fișierul trebuie să conțină două linii. Prima linie conține un întreg: numărul N al obiectelor. A doua linie conține numere întregi de la 1 la N într-o ordine oarecare, al i-lea întreg reprezentând puterea obiectului etichetat cu i.

Exemplu

DEVICE.IN

5

2 5 4 3 1

Fișierul descrie o intrare cu 5 obiecte și puterile lor.

Etichetă : 1 2 3 4 5

Putere : 2 5 4 3 1

O secvență este formată din următoarele 5 apeluri :

1. GetN (în Pascal) sau GetN() (în C/C++) – returnează 5
2. Med3(1,2,3) – returnează 3
3. Med3(3,4,1) – returnează 4
4. Med3(4,2,5) – returnează 4
5. Answer(4).

Restricții

Numărul N al obiectelor este impar și $5 \leq N \leq 1499$

Orice etichetă i satisface relația $1 \leq i \leq N$

Orice putere satisface relația $1 \leq Y \leq N$; puterile sunt distincte

Numele bibliotecii Pascal este DEVICE.TPU

Declarațiile subprogramelor Pascal sunt :

function GetN : Integer;

function Med3(x,y,z:Integer) : Integer;

procedure Answer(m:integer);

Numele bibliotecilor C/C++: DEVICE.H, DEVICE.OBJ (folosiți modelul de memorie large)

Header-ele funcțiilor C/C++ sunt :

int GetN(void);

int Med3(int x, int y, int z);

void Answer(int n);

La o rulare sunt permise cel mult 7777 de apeluri ale subprogramului Med3

Programul dumneavoastră nu trebuie să conțină instrucțiuni explicite de citire/scriere din/în vreun fișier; citirile/scrierile se realizează prin intermediul subprogramelor din bibliotecă.

9. Temă

1. Citiți enunțul și rezolvarea problemei 13 de pe site-ul <http://olympiads.win.tue.nl/ioi>
2. Rezolvați ecuația $y^3 = x$, unde $x, y \in S_n$ ($n \leq 1000$) (problema 12)
3. Rezolvați problema 10 (problema lui Mihai Stroe de la Baraj)
4. Găsiți p^n pentru n mare (problema 2)

BIBLIOGRAFIE

1. I. Tomescu – Probleme de combinatorică și teoria grafurilor, Ed. Didactică și Pedagogică - București 1981
2. T. Cormen, C. Leiserson, R. Rivest – Introducere în algoritmi, Ed. Computer Libris Agora – Cluj Napoca 2000
3. A. P. Domoread – Jocuri și probleme distractive de matematică, Ed. Didactică și Pedagogică - București 1965
4. D. Knuth – Arta programării calculatoarelor vol.1, Ed. Tehnică – București 1977
5. V. Mitrana – Provocarea algoritmilor, Editura Agni – București 1994
6. C. Năstăsescu, C. Niță, M. Brandiburu, D. Joița – Exerciții și probleme de algebră, Ed. Didactică și Pedagogică - București 1983
7. I. Tomescu – Introducere în combinatorică, Ed. Tehnică - București 1972
8. Seria GInfo numerele 8/1 – 10/8, 1998-2001.