



OLIMPIADA NAȚIONALĂ DE INFORMATICĂ BRĂILA 26 APRILIE – 03 MAI 2002

Clasele XI-XII

Sursa: `DECOD.pas`, `DECOD.c`, `DECOD.cpp`

Ieșire: `DECOD.out`

DECOD

Serviciul Român de Informații (SRI) a dat de urma unei binecunoscute și periculoase organizații teroriste, care își are sediul pe teritoriul țării noastre. Folosind cei mai pricepuți și mai bine antrenați spioni și ofițeri, SRI a reușit să identifice computerul principal al organizației teroriste. Dacă va reuși să acceseze informațiile din acest computer, SRI îi va putea aresta pe toți membrii organizației și va asigura (în continuare) pacea mondială. Singura problemă este spargerea codului de acces. Tot ceea ce se știe despre acest cod este că el este reprezentat de către o permutare de lungime N . Specialiștii SRI au încercat diverse metode de a descoperi codul, însă tot ceea ce au reușit să obțină este un program care, transmitându-i-se ca parametru o permutare de lungime N , specifică în câte poziții coincid această permutare și codul de acces. Din păcate, șefii nu cred în utilitatea acestui program.

Cerință

Scrieți un program care, folosind programul-ajutător (reprezentat sub forma unui modul), să determine codul de acces în computerul teroriștilor.

Date de intrare

Programul dumneavoastră nu va citi date din vreun fișier de intrare. El va apela întâi funcția **GetN** a modului **PROG**, care îi va returna valoarea N – numărul de elemente ale permutării ce trebuie descoperită.

Apoi va apela numai funcția **Check**, căreia îi va fi transmisă ca parametru, de fiecare dată, o permutare de lungime N . Această funcție va returna numărul de poziții în care permutarea transmisă ca parametru coincide cu permutarea ce trebuie descoperită. Programul dumneavoastră trebuie ca, după un număr finit de apelări ale funcției **Check**, să descopere permutarea căutată.

Date de ieșire

Programul dumneavoastră va trebui să tipărească în fișierul **DECOD.OUT** o permutare de lungime N . Toate cele N elemente vor fi tipărite pe prima linie a fișierului, fiind separate de câte un spațiu.

Restricții:

- $5 \leq N \leq 256$

Instrucțiuni pentru programatorii în C/C++

Veți avea la dispoziție header-ul **prog.h**, pe care va trebui să-l includeți în sursa dumneavoastră. Acest fișier declară următoarele funcții:

```
int GetN ( void )
```

```
int Check ( int p[256] )
```

Funcția **Check** are ca parametru un vector cu elemente întregi, care reprezintă o permutare. Ea va evalua această permutare și va returna una dintre următoarele valori:

- 1 dacă primele N elemente întregi (începând cu poziția 0) ale vectorului transmis ca parametru nu constituie o permutare de lungime N
- 0 – N numărul de poziții în care coincide permutarea de lungime N transmisă ca parametru, cu permutarea ce trebuie descoperită



Clasele XI-XII

Instrucțiuni pentru programatorii în PASCAL

Modulul **PROG** este implementat sub forma unit-ului **PROG**. În acest unit sunt definite următoarele tipuri și funcții, care vor fi folosite de către programul dumneavoastră:

- tipuri:
perm = array [1..256] of integer;
- funcții:
function GetN : integer;
function Check (var permut : perm) : integer;

Funcția **Check** are ca parametru un vector de tipul **perm**. Ea va evalua aceasta permutare și va returna una din următoarele valori:

- 1 dacă primele N elemente întregi ale vectorului transmis ca parametru nu constituie o permutare de lungime N
- 0 - N numărul de poziții în care coincide permutarea de lungime N transmisă ca parametru, cu permutarea ce trebuie descoperită

Exemplu de apel

<pre>/* C/C++ */ int p[256]; ... tmp = Check(p);</pre>	<pre>{ Pascal } var p: perm; ... tmp := Check(p);</pre>
--	---

Pentru a vă testa programul, creați un fișier numit **DECOD.IN**. Pe prima linie a acestui fișier scrieți valoarea N. Pe a doua linie scrieți o permutare de lungime N, având elementele separate prin spațiu. Modulul **PROG** va citi această permutare din fișierul **DECOD.IN** (din directorul curent) și o va considera permutarea ce trebuie descoperită.

Exemplu

DECOD.IN

```
5  
2 1 3 4 5
```

O serie posibilă de apeluri ale funcțiilor **GetN** și **Check** este următoarea:

- Se apelează funcția **GetN** și se returnează valoarea 5
- Se apelează funcția **Check** cu permutarea (1 2 3 4 5) și se returnează valoarea 3
- Se apelează funcția **Check** cu permutarea (3 2 1 4 5) și se returnează valoarea 2
- Se apelează funcția **Check** cu permutarea (2 1 3 4 5) și se returnează valoarea 5
- Se tipărește în fișierul **DECOD.OUT** permutarea 2 1 3 4 5. Fișierul **DECOD.OUT** va arăta astfel:
2 1 3 4 5 - pe prima linie a fișierului

Timp maxim de executare: 1 secundă/test

Notă:

Pe calculatorul dumneavoastră va fi instalat o versiune a modulului **PROG** în directorul `c:\decod\`, pe care o puteți folosi pentru testare. Înainte de evaluare, programul dumneavoastră va fi compilat folosind o altă versiune a modulului **PROG**. Această versiune nu va necesita un timp mai mare pentru executarea funcțiilor puse la dispoziție decât varianta de pe calculatoarele dumneavoastră.