## 1. Feladat –biperm 100 pont

Egy **n** természetes szám ismeretében tekintsük az összes nem nulla, **n**-nél kisebb vagy egyenlő természetes szűmot, mindegyiket kétszer: **1**, **1**, **2**, **2**, **3**, **3**, ... , **n**, **n**. Ezeket a számokat tetszőlegesen összekeverjük és két sorba rakjuk, minden sorba **n** elemet. Egy ilyen adatstruktúrát *bipermutációnak* hívunk. Az **1.**, **2.** és **3.** ábrán egy-egy bipermutációt látunk **n**=5-re.

Egy bipermutáció *tökéletes*, ha mindkét sora egy-egy permutáció (lásd a **2.** és **3.** ábrákat).

Egy *csere a* **p** *pozíción* úgy van értelmezve, mint a **p**-ik oszlopban levő elemek felcserélése. Az alábbi példák esetében, a **2**-ik, illetve a **3**-ik ábrán levő tökéletes permutációkat az első ábrán levő bipermutációból nyertük miután egy műveletet alkalmaztunk a **2**-ik pozíción, illetve **4** műveletet alkalmaztunk az **1**, **2**, **4** és **5**-ik pozíciókon.

| | | | | |
|---|---|---|---|---|
| 1 | 5 | 5 | 3 | 4 |
| 3 | 2 | 2 | 4 | 1 |

**1.** *ábra*

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 5 | 3 | 4 |
| 3 | 5 | 2 | 4 | 1 |

**2.** *ábra*

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 1 |
| 1 | 5 | 2 | 3 | 4 |

**3.** *ábra*

**Követelmény**

Ismerve egy bipermutációt, határozzátok meg:

- az egymástól különböző bipermutációk számát, amiket cserékkel tudunk létrehozni;
- a cserék minimálás számát, melyek alkalmazása során tökéletes permutációt nyerünk;
- egy tökéletes bipermutációt, amelyet az eredetileg beolvasott bipermutációból nyerünk.

**Bemeneti adatok**

A `biperm.in` bemeneti állomány első sora tartalmazza **n** értékét. A következő két sor mindegyike **n** elemet tartalmaz szóközzel elválasztva, ami egy bipermutációt jelöl.

**Kimeneti adatok**

A `biperm.out` kimeneti állomány tartalma:

- az első sorban két természetes szám, szóközzel elválasztva. Az első szám az egymástól különböző bipermutációk száma, a második pedig a cserék minimális száma, amivel tökéletes permutációt nyerünk.
- a következő két sorban rendre **n** természetes szám egy-egy szóközzel elválasztva, amely egy tökéletes bipermutáció, amit az eredeti bipermutációból nyertünk.

**Megszorítások és pontosítások**

- $2 < n \leq 10\,000$;
- az egymástól különböző bipermutációk számának helyes kiszámítása **30%** -ot ér az összpontból;
- a cserék minimális számának helyes kiszámítása **10%** -ot ér az összpontból;
- egy tökéletes bipermutáció **60%**-ot ér az összpontból. Több megoldás is létezhet, bármelyik helyes megoldás el van fogadva;
- a különböző bipermutációk száma egyetlen teszt esetében sem haladja meg a 2 000 000 000-t.
- egy helyes válasz csak akkor lesz elfogadva, ha az előtte levő válaszok léteznek, függetlenül ezek helyességétől;
- a tesztek **40%**-a esetében $n \leq 20$
- a tesztek **40%**-a esetében $20 < n \leq 400$
- a tesztek **20%**-a esetében $400 < n \leq 10\,000$

**Példa**

| biperm.in | biperm.out | Magyarázatok |
|---|---|---|
| 5<br>1 5 5 3 4<br>3 2 2 4 1 | 4 1<br>1 2 5 3 4<br>3 5 2 4 1 | Van **4** tökéletes permutációnk. A cserék minimális száma **1** és két megoldás is létezik minimális cserével:<br>1 2 5 3 4 1 5 2 3 4<br>3 5 2 4 1 és 3 2 5 4 1<br>A másik két megoldás, amit nem minimális cserével kapunk meg:<br>3 2 5 4 1 3 5 2 4 1<br>1 5 2 3 4 és 1 2 5 3 4<br>A harmadik követelményt a fenti **4** megoldás bármelyike teljesíti. |

Maximális futási idő: **1** másodperc

Maximális memória: **128** MB amiből a verem mérete **64** MB

A forráskód maximális mérete: **10** KB

**Problema 1 – biperm** **100 puncte**

Pentru un număr natural nenul **n**, să considerăm toate numerele naturale nenule mai mici sau egale cu **n**, luând fiecare număr de câte două ori: **1**, **1**, **2**, **2**, **3**, **3**, ... , **n**, **n**. Aceste numere le amestecăm aleator, şi le aranjăm pe două linii a câte **n** elemente. Structura astfel obținută o vom numi o *bipermutare*. În *figurile* **1**, **2** şi **3** avem câte un exemplu de bipermutare pentru **n=5**.

O bipermutare este *perfectă*, dacă ambele linii ale structurii reprezintă câte o permutare (vezi *figurile* **2** *şi* **3**).

Prin *mutare pe poziția* **p**, înțelegem interschimbarea elementelor de pe aceeaşi coloană **p**. În exemplele de mai jos, bipermutarea perfectă din figura **2** s-a obținut din bipermutarea din figura **1**, aplicând o mutare pe poziția **2**. Bipermutarea perfectă din figura **3** s-a obținut din bipermutarea din figura **1**, aplicând mutări pe pozițiile **1**, **2**, **4** şi **5**.

| | | | | |
|---|---|---|---|---|
| 1 | 5 | 5 | 3 | 4 |
| 3 | 2 | 2 | 4 | 1 |

*Figura* **1**

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 5 | 3 | 4 |
| 3 | 5 | 2 | 4 | 1 |

*Figura* **2**

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 1 |
| 1 | 5 | 2 | 3 | 4 |

*Figura* **3**

**Cerințe**

Cunoscând o bipermutare, determinați:

- numărul bipermutărilor perfecte distincte ce se pot obține prin mutări;
- numărul minim de mutări prin care se poate obține o bipermutare perfectă;
- o bipermutare perfectă obținută din bipermutarea inițială.

**Date de intrare**

Fişierul de intrare `biperm.in` conține pe prima linie valoarea lui **n**. Următoarele două linii conțin, fiecare, câte `n` elemente separate prin câte un spațiu, formând o bipermutare.

**Date de ieşire**

Fişierul de ieşire `biperm.out` va conține:

- pe prima linie două numere naturale separate printr-un spațiu, reprezentând numărul bipermutărilor perfecte distincte ce se pot obține din bipermutarea dată, respectiv numărul minim de mutări prin care se poate obține o bipermutare perfectă;
- pe următoarele două linii se vor tipări câte **n** numere separate prin spațiu, reprezentând o bipermutare perfectă obținută din bipermutarea dată.

**Restricții şi precizări**

- **2 < n ≤ 10 000**;
- calculul corect al numărului bipermutărilor perfecte distincte valorează **30%** din punctaj;
- calculul corect al numărului minim de mutări valorează **10%** din punctaj;
- tipărirea unei bipermutări perfecte valorează **60%** din punctaj. Pot exista mai multe soluții, se va admite orice soluție corectă;
- se garantează că numărul bipermutărilor perfecte distincte nu depăşeşte **2 000 000 000** pentru niciun test.
- acordarea punctajului la un răspuns corect este condiționată de existența răspunsurilor anterioare, indiferent de corectitudinea lor;
- pentru **40%** din teste **n ≤ 20**
- pentru **40%** din teste **20 < n ≤ 400**
- pentru **20%** din teste **400 < n ≤ 10 000**

**Exemplu**

| biperm.in | biperm.out | Explicații |
|---|---|---|
| 5<br>1 5 5 3 4<br>3 2 2 4 1 | 4 1<br>1 2 5 3 4<br>3 5 2 4 1 | Sunt **4** permutări perfecte. Numărul minim de mutări este **1** şi există două soluții cu număr minim de mutări:<br>1 2 5 3 4 şi 1 5 2 3 4<br>3 5 2 4 1 şi 3 2 5 4 1<br>Celelalte două soluții, ce nu se obțin din număr minim de mutări sunt:<br>3 2 5 4 1 şi 3 5 2 4 1<br>1 5 2 3 4 şi 1 2 5 3 4<br>Pentru a treia cerință oricare dintre cele **4** soluții este acceptată. |

Timp maxim de execuție per test: **1** secundă
Total memorie disponibilă: **128** MB din care **64** MB pentru stivă
Dimensiunea maximă a sursei : **10** KB

## 2. feladat - subsecvente 100 pont

Legyen **n** egy természetes szám és **M**={$S_1$,$S_2$,…,$S_n$} egy nem üres, karaktersorokból álló halmaz, ahol minden karaktersorban az elemek az { 'a', 'b' } halmazból vannak véve. |$S_k$|-val jelöljük az $S_k$ sor karaktereinek számát, vagy másképp megnevezve, az $S_k$ hosszát. Az $S_k$ egy részsora úgy van értelmezve mint az **i**, **i+1**, .., **j-1**, **j** pozíciókon helyezkedő karakterek. Jelöljük ezt a részsort $S_k$**[i:j]**-vel. Ezek szerint, ha $S_k$ = 'abbbaababa', akkor $S_k$**[3:6]** = 'bbaa' vagyis a kiemelt részsor: 'ab**bbaa**baba'.

### Követelmény
Adott egy **M** halmaz, számítsuk ki egy részsor maximális hosszát, amelyik **M** minden karaktersorában megtalálható.

### Bemeneti adatok
A subsecvente.in bemeneti állomány első sorában található egy **n** természetes szám, amely az **M** halmaz kardinális számával egyenlő. A következő **n** sor mindegyike az **M** halmaz egy-egy karaktersorát tartalmazza.

### Kimeneti adatok
A subsecvente.out kimeneti állományba egyetlen számot kell beírni, a megtalált részsor hosszát.

### Megszorítások
- **1 < n < 5.**
- Ha |**S**| = |$S_1$| + |$S_2$| + … + |$S_n$|, akkor |**S**| < **50 001**.
- Garantáltan létezik minden esetben megoldás.
- A megoldás garantéltan nem nagyobb mint **60**.
- A tesztek **30%**-ra: |**S**| < **101**.
- A tesztek **55%**-ra: |**S**| < **3 501**.
- A tesztek **80%**-ra: |**S**| < **10 001**.

### Példa

| subsecvente.in | subsecvente.out | **Magyarázat** |
|---|---|---|
| 4<br>abbabaaaaabb<br>aaaababab<br>bbbbaaaab<br>aaaaaaabaaab | 5 | Egy maximális hosszúságú közös részsor hossza 5.<br><br>A mi példánkban az 5 hosszúságú közös részsor aaaab:<br>abbaba**aaaab**b, **aaaab**abab, bbbb**aaaab**, aaa**aaaab**aaab. |

Maximális futási idő: **0.6s**.
Maximális memória: **128MB** amiből **64MB** a verem mérete
A forráskód maximális mérete: **20KB**

## Problema 2 - subsecvente 100 puncte

Fie **n** un număr natural și **M**={$S_1, S_2, \ldots, S_n$} o mulțime de șiruri de caractere nevide. Fie $S_k$ un șir de caractere din **M**. Atunci, orice caracter al lui $S_k$ aparține mulțimii { 'a', 'b' }. Notăm prin $|S_k|$ numărul caracterelor șirului $S_k$ sau, echivalent, lungimea sa. O subsecvență $S_k[i:j]$ a lui $S_k$ este formată din caracterele situate pe pozițiile consecutive **i**, **i+1**, .., **j**. Astfel, dacă $S_k$ = 'abbbaababa', atunci $S_k[3:6]$ = 'bbaa' sau subsecvența evidențiată: 'ab**bbaa**baba'.

### Cerință
Fiind dată o mulțime **M**, se cere să se determine lungimea maximă a unei subsecvențe care se găsește în toate șirurile din **M**.

### Date de intrare
Pe prima linie a fișierului de intrare subsecvente.in se găsește un număr natural **n** egal cu cardinalul mulțimii **M**. Pe fiecare din următoarele **n** linii se găsește câte un șir din mulțimea **M**.

### Date de ieșire
Pe prima linie a fișierului de ieșire subsecvente.out se va scrie un singur număr natural egal cu lungimea subsecvenței găsite.

### Restricții
- **1** < **n** < **5**
- Dacă $|S| = |S_1| + |S_2| + \ldots + |S_n|$, atunci $|S|$ < **50 001**
- Se garantează că va exista întotdeauna soluție
- Se garantează că rezultatul nu va depăși **60**
- Pentru **30%** din teste: $|S|$ < **101**
- Pentru **55%** din teste: $|S|$ < **3 501**
- Pentru **80%** din teste: $|S|$ < **10 001**

### Exemplu

| subsecvente.in | subsecvente.out | Explicație |
|---|---|---|
| 4<br>abbabaaaaabb<br>aaaababab<br>bbbbaaaab<br>aaaaaaabaaab | 5 | Lungimea unei subsecvențe comune de lungime maximă este 5.<br><br>În exemplu subsecvența comună de lungime 5 este aaaab:<br>abbaba**aaaab**b, **aaaab**abab,<br>bbbb**aaaab**, aaa**aaaab**aaab. |

Timp de execuție per test: **0.6s**
Memorie per test: **128MB** din care **64MB** pentru stivă
Dimensiunea maximă a sursei: **20KB**