

# ХІІІ МЕЖДУНАРОДНА ОЛИМПИАДА ПО ИНФОРМАТИКА

## Контролно състезание, 10.06.2001г.

### Задача 1. Пермутации

Игра се играе със зададена пермутация на целите числа от 1 до  $N$  ( $5 \leq N \leq 1000$ ). На всеки ход на играта може да се вземе блок от произволен ненулев брой числа, разположени последователно в пермутацията и да се премести на произволно място в пермутацията (включително преди първия и след последния елемент), без да се променя реда на числата в блока. Така се получава нова пермутация на числата от 1 до  $N$ , с която може да се направи същия ход и т.н. Целта е да се получи пермутацията  $1, 2, \dots, N$  с колкото може по-малко ходове. В първия ред на входният файл `PERM.INP` ще е зададено числото  $N$ , а следващите  $N$  реда ще съдържат последователните елементи на пермутацията.

Напишете програма `PERM.EXE`, която да решава поставената задача, като за целта комуникира със зададен програмен модул. За пишешите на C модулът се нарича `modul.obj` и за връзка с него е необходимо да включите в програмата си редът `#include "m.h"`, съдържащ декларации на функциите:

`unsigned start()` – инициализира работата на модула и връща уникален ключ `key` за комуникация между него и програмата Ви;

`void move(unsigned key, unsigned L, unsigned M, unsigned K)` – регистрира преместване направено от програмата Ви на елементите намиращи се от позиция  $L$  до позиция  $M$  в пермутацията след елемента намиращ се в позиция  $K$ . За да бъде изпълнена тази функция е необходимо да са изпълнени условията  $1 \leq L \leq M \leq N$ , а  $K < L$  или  $K > M$ . Ако  $K=0$  блокът от числа ще бъде поставен в началото на текущата пермутация. За пишешите на Pascal.....

Ако поредното извикване на `move` завършва подреждането на пермутацията, тя отпечатва на екрана броя на направените ходове и прекратява изпълнението на програмата.

Оценката на работата на програмата ще стане на базата на Алгоритъм, който не винаги прави оптималния брой ходове. Ако програмата ви подреди пермутацията с брой ходове по-малък или равен на тези с което извършва подреждането Алгоритъмът, ще получите пълния брой точки (10). За всяко подреждане, с повече ходове от тези, които извършва Алгоритъмът ще получите част от пълния брой точки, пропорционален на отдалечеността на броя ходове на вашата програма от броя ходове направени от алгоритъма.

### Пример

Ако  $N=5$  и пермутацията е 3 5 1 4 2, тогава с извикванията

```
move(key, 3, 3, 0)
```

```
move(key, 5, 5, 1)
```

```
move(key, 5, 5, 3)
```

пермутацията ще бъде подредена за 3 хода, а с

```
move(key, 4, 4, 1)
```

```
move(key, 1, 3, 5)
```

за 2 хода.